University of Alberta

The Streetprint Engine

by

Matthew Scott Ogle ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the

requirements for the degree of Master of Arts

Department of English

Edmonton, Alberta
Fall 2004

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-95647-4
Our file Notre référence
ISBN: 0-612-95647-4

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canadä

Table of Contents

Introduction 1

Chapter 1

The Streetprint Engine: A Digital Archive Bildungsroman 8

Think Before You Code 8
Code Before You Think (Too Much) 13
Programming Politics 18
A Lot of People Could Use This Thing 20
Streetprint: The Next Generation 27
OOP, I Did It Again 35
Streetprint 2.0 43
Not Invented Here, But You'll Like It: Streetprint 2.1 48
Arkives and the Future of Streetprint 55

Chapter 2

Hackers, Humanists, Intellectuals: Lessons from the Open Source Revolution 59

Works Cited 80

Appendix A 84

Appendix B 138

Introduction

> We are thus entering a period when the entirety of our received cultural archive of materials, not least of all our books and manuscripts, will have to be [digitally] reconceived...information scientists and systems engineers will be (already are) much involved with these changes. But it is the literary scholar, the musicologist, the art historian, etc. who have the most intimate understanding of our inherited cultural materials.
>
> —Jerome McGann, *radiant textuality*, 169

In the spring of 2002, I had just completed my undergraduate arts degree at the University of Alberta. I convocated with an unusual degree in English Literature (major) and Computing Science (minor), a combination which—despite a decades-old tradition of computing in the humanities—still managed to elicit confused stares from colleagues in both disciplines, not to mention the university administration. My coursework proved to be so enjoyable that I hardly noticed the disjunction between my two areas of study. Occasionally, I'd become inspired and try to persuade skeptics that the gap between tech and text was a largely imaginary, unproductive fissure. Though my convictions were solid, my converts were few.

Things took a turn for the better when I was hired by Gary Kelly, the University of Alberta's newly-appointed Canada Research Chair in Literature and Language in Society. Kelly was in the process of setting up a multimedia computing studio intended for humanities researchers, funded primarily by the Canada Foundation for Innovation (CFI). Ostensibly hired for the four months of summer, I became the CRC Studio's first employee. And that's when I discovered that Kelly, an expert in romanticist women's writing and popular print, had a problem. Furthermore, he hoped digital technology could provide the solution. "Matt," he explained, "I need a website."

Since the early 1980s, Kelly had been collecting street literature—ballads, chapbooks, political pamphlets, and so on—from late eighteenth- and early nineteenth-century Britain. This private collection had grown to several hundred items, many of which (by virtue of their age and the frugality of their publishers) were extremely fragile. Sharing this valuable body of material with other researchers was problematic due to the ephemeral nature of the texts. Teaching with the material was also difficult, since providing students with direct access to the physical texts risked turning them into dust in fairly short order. In 2001, Kelly made an attempt to teach some of the collection's material using versions set in modern-day type, edited by graduate research assistants into a series of Microsoft Word documents. This experiment was a testament to the signifying power of what Jerome McGann terms 'bibliographical encoding.' While students now had access to the linguistic elements of the texts, they could not experience the texts' many significant visual features, including the cheapness of the paper, the way the text had been cut or folded, the typefaces chosen, the woodcut illustrations, the marginalia, and the countless other bibliographical codes which spoke volumes about each text's history, publication, dissemination, circulation, and use; in short, its *context*. A crucial part of what made the collection compelling had been lost in translation. Clearly, an opportunity to more fully engage students and other researchers was being missed.

Gary Kelly hoped his popular print research problem could be solved by moving his collection online and into the public sphere of the internet, and this hope was hardly surprising. When the internet hit its stride in the mid-1990s, it seemed for a while that websites (and their digital kin) could present a solution to *every* problem, ushering in a democratized, populist paradise in the process. Computers had developed impressive imaging and multimedia capabilities, and the burgeoning information superhighway promised to connect and empower like never before. Freedom to share and communicate, we were told, was hard-coded into the infrastructure of the net itself, an inevitable product of going digital. "The information superhighway," gushed MIT Media Lab founder Nicholas Negroponte in his 1995 book *Being Digital*, "is about the global movement of weightless bits at the speed of light" (12). By 2002, however, visions of this impending digital paradise (within both the academy and the broader community) had been tempered by the more pragmatic realities of the dot-com bust, among other factors. Not that Negroponte and the other digital prophets had it wrong; while digital bits are weightless, figuring out what to *do* with those bits (and how to structure them) presents a weighty problem indeed. And as Harvard law professor Lawrence Lessig reminds us, nothing in cyberspace is natural or automatic, but (like every other space) is constructed: "architectures constitute cyberspace; these architectures are varied; they variously embed political values" (*Code* 217). By 2002, cyberspace was becoming increasingly recognized as a complicated, political, and *architected* space that is subject to change—not unlike a university campus.

Up until the summer of 2002, the digital architectures available to this particular university professor with an interest in popular print had failed to meet his needs. The gap between Gary Kelly's expertise in British street literature and the technical expertise required to mount a useful digital collection was too great, and he was hoping that the newly-created CRC Studio could help bridge that gap. As the studio's first employee working on the studio's first project, I began researching the problem. "You don't just need a website," I said after examining Kelly's collection for a few weeks. "You need a database." And the database needed a name, so we called it *Streetprint*.

My reason for proposing a database-driven website was simple: to borrow McGann's phrase, Gary Kelly was the researcher with "the most intimate understanding" of these particular cultural materials. If I could apply my understanding of literature and computing to the construction of a database which then powered a website showcasing Kelly's popular print, I could build easy-to-use tools giving Kelly control over his own collection. Once I left the CRC Studio at the end of the summer, the theory went, control over the content and even organization of the Streetprint database would be in the hands of Kelly, the cultural expert, and not Matthew Ogle, the technician.

In order to remain as true as possible to the spirit of the collection we hoped to digitize, two additional decisions were made. First, Streetprint would use *images* of each text as a structuring principle, as opposed to an encoded textual representation of each item. Given our needs, this decision was not a difficult one. I had many ideas about how to ensure that an image-oriented database would still contain enough textual metadata to make it easily searchable and browsable, and I also agreed in the main with McGann's assertion that current standards for text encoding (such as XML and SGML) were "not only difficult and time-consuming to implement, but [that their] hierarchical principles and other design characteristics set permanent and unacceptable limits on [their] usefulness with arts and humanities materials" (*radiant textuality* 17). The CRC Studio had neither the time nor the resources to launch a text encoding initiative which would essentially strip the collection's texts of their valuable paratexts and visual characteristics. Instead, it was decided that Streetprint would attempt to mimic, as closely as possible, the experience of sitting down with the actual texts themselves. I also hoped that by focusing on images, we could create a user experience which encouraged free-form browsing alongside more "traditional" support for searchers who already knew what they were looking for.

A second important decision came once I began working in earnest on creating the website which would interact with Streetprint's underlying database. I quickly realized that, despite my experience in both computing and literary studies, my desire to do justice to the collection's textual and visual integrity was quickly exceeding my competence in one crucial area: design. In June of 2002, recent University of Alberta Bachelor of Design graduate Chris Govias was hired to work on Streetprint, becoming the CRC Studio's first in-house designer in the process. Govias' arrival transformed the entire project, making us realize something which has since become a central tenet of the CRC Studio: reporting research is, first and foremost, a problem of *design*. Just as we were unwilling to strip the visual elements from our 200-year old popular print, we quickly came to understand that design—incorporating visual design, information design, typography, and more—is not simply window dressing which can be applied at will to a nearly-completed project. It is central to the development of any research project, and especially to a project which hopes to effectively communicate research in the wonderfully (and notoriously) flexible digital medium. By the end of June 2002, it was clear that Streetprint was going to benefit tremendously from the fortuitous early addition of a designer to its development team.

Almost every research project has a "eureka" moment, a sudden realization which forever changes the course and scope of the task at hand. Streetprint's eureka came in July 2002, and it is the reason that I didn't leave the CRC Studio as planned in September 2002, the reason that I am here, writing a master's thesis about Streetprint, nearly two years later. July 2002 found me in Gary

Kelly's office, excitedly explaining how well Streetprint was developing into an ideal solution to his popular print research problem. "You know," Kelly began, "we're not the only ones with this problem. A lot of people could use this thing." This seemingly simple observation had far-reaching implications. What if the solution we were developing for a particular problem and a particular collection could instead be applied to a *variety* of research collections in popular print and beyond? What if we could create a tool useful to literary historians, collectors, musicologists, and the countless others who are not technical experts but whose expertise is indispensable in digitally preserving our inherited cultural archives? What if these people could use the internet to reach out to new communities of researchers? What if we could provide this software to researchers with even the smallest of technological and financial resources? What kind of models existed for this sort of software? Should English majors really write code? And what should we call this thing?

We called it the *Streetprint Engine*. This thesis tells the story of the past two years, the story of how Streetprint grew from a single research website into an evolving, open source software engine which powers more than a dozen different online collections. Part genesis narrative and part humanities computing case study, this work examines the surprising theoretical junctures between tech and text, hacker and humanist. It reflects upon both the successes and failures of the Streetprint Engine, hopefully pointing toward a new direction for digital archives—and humanities computing—along the way.

Chapter 1 is entitled "The Streetprint Engine: A Digital Archive Bildungsroman." It expands upon the initial events leading up to the birth of the Streetprint Engine, and traces the evolution of the project between 2002 and 2004. Theorist Ellen Rose has remarked on the representation of computer software as an anonymous, autonomous entity, explaining that "to make software seem digitally perfect all evidence of human involvement, all traces of the relations of production, must be eliminated" (62). Chapter 1 attempts to reverse that tendency, making explicit the very human decisions made by the Streetprint development team during the creation of our digital infrastructures and user documentation, and during our continuous attempts to find connections between literary and computing theory and practice. The chapter concludes with a look at a new, upcoming version of the Streetprint Engine and the challenges which lie ahead.

Chapter 2, "Hackers, Humanists, Intellectuals: Lessons from the Open Source Revolution," traces the history of the free software / open source software movement(s) and highlights their implications—philosophic and pragmatic—for digital archives and for the humanist researcher in general. In doing so, this chapter provides a broader context for the Streetprint Engine, situating the project within a larger framework of evolving technological, literary, and social movements.

Following these two primary chapters and a bibliography is a series of appendices. Appendix A contains a series of planning illustrations, technical specifications, and screen captures related to the development of the Streetprint Engine, from its initial version (1.0) through to our upcoming 3.x development version. Included are the complete database specifications for each successive version of the Streetprint Engine, as well a set of UML (Unified Modeling Language) planning diagrams used during the development of the Streetprint Engine version 2.0. Appendix B includes the complete text of the user manual for the current version of the Streetprint Engine.

Chapter 1

The Streetprint Engine: A Digital Archive Bildungsroman

> Any ethics or morals or second thoughts, any questions or muddles or exceptions, all dissolve...if I just sit here and code, you think, I can make something run. When the humans come back to talk changes, I can just run the program. Show them: Here. Look at this...whatever you might say, whatever the consequences, all you have are words and what I have is this, this thing I've built, this operational system. Talk all you want, but this thing here: it *works*.
>
> — Ellen Ullman, *Close to the Machine*, 24

> Software production is also typically a team effort, yet most software programs are entirely anonymous: they offer no scrolling credits to remind us that a computer program, like a film, is authored. We are simply not meant to consider the processes by which designers, programmers, graphic artists, and others have worked together to transform the ephemera of ideas into the equally ephemeral virtual reality of computer software...
>
> — Ellen Rose, *User Error*, 62

> ALL TEXTS ARE PRODUCED OVER TIME AND UNDER VARYING CIRCUMSTANCES...TEXTS MUST EMPHASIZE THEIR RELATIONS, AND THEIR RELATIVITIES.
>
> —Jerome McGann, *The Textual Condition*, 93

Think Before You Code

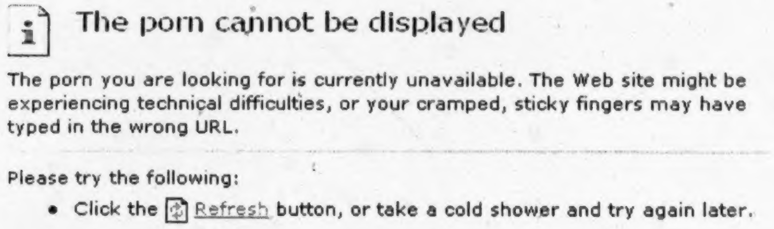
When I first began developing Streetprint, my chief fear was that I too—like my predecessors, who had transcribed texts from the collection into Microsoft Word documents—would irreparably harm the collection's integrity during the transition from (chap)books to bits. Despite my love for (and faith in) the digital medium, I worried that the gulf between centuries might be too great, that the CRC Studio's resources might be too meager, that the totalizing, "make it work" logic of databases and programming code might serve to erase the relations and relativities so important to Gary Kelly's archive of popular print. In an attempt to ease these fears before making the leap into cyberspace, I decided to research the history of the texts I was hoping to digitize. What I found were a number of surprising, reassuring sonorities between theories of popular print and popular conceptions of our new digital age.

"Sub-literature," writes literary historian Leslie Shephard, "was linked to the fierce energies of the crowds that created history, rather than to the culture of its rulers or manipulators" (14). In his fiery 1996 "Declaration of the Independence of Cyberspace," John Perry Barlow, co-founder of the Electronic Frontier Foundation (EFF), invokes a similar 'fierce energy.' Addressed to the "governments of the Industrial World, you weary giants of flesh and steel," Barlow makes it clear that the discourse of cyberspace similarly belongs not to the rulers, but to the digital crowds: "you are not welcome among us...you have no sovereignty where we gather...you have not engaged in our great and gathering conversation." (1) While Barlow's epic proclamation may highlight a relatively tenuous philosophical similarity between the values of street literature and cyberspace, more concrete and prosaic resemblances also emerged during my initial research.

Ballad sheets—which comprise roughly half of Kelly's collection—were a form of popular print which, as Shephard explains, "provided news, diversion, inspiration, fantasy, and political stimulus" (14) to their readers. This sounded a lot like the internet to me; reading on, I discovered that these texts not only communicated, but *connected*: "ballads reached the country side by way of country chapmen...they were often the only link had between themselves and the outside world" (88). Sold for a penny or half-penny by these singing chapmen, ballad sheets circulated far and wide amongst their literate and semi-literate audience, changing hands as often as a humorous email forward or web link. Bawdy parodies were sometimes sold right alongside the original ballads. "A Parody on Patty Kavannah[1]", one ballad sheet from Kelly's collection, turns the romantic rhymes of the original ("Shall we meet ere evening's grey, / Hey Patty, pretty Patty?") into something slightly more gritty:

> Will yon meet in Gray's-Inn-Jane?
> Say Becky, dirty Becky,
> In the mud and in the rain,
> Dirty Becky, tell me.
> When I'm there with my mud-cart,
> To take some grub before we part,
> And eat off a sheep's heart,
> Cinder-sifting Becky.

The visual elements of this parody enhance the comic effect, since the printer even uses the exact same woodcut illustration which accompanies the original ballad, "Lovely Patty Kavannah[2]." Parody remains wildly popular on the internet, and the technique—especially the attention to visual detail—is largely the same, from pointed recreations of CNN.com[3] ("Bush to Saddam – Bite Me!") to the many humorous reworkings of the well-known "404 Page Not Found" error message:



The porn cannot be displayed

The porn you are looking for is currently unavailable. The Web site might be experiencing technical difficulties, or your cramped, sticky fingers may have typed in the wrong URL.

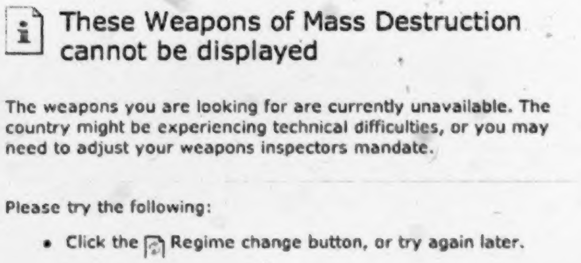
Please try the following:

- Click the Refresh button, or take a cold shower and try again later.

Figure 1-1, from http://my.core.com/-oldgrendel/cannot_find_porn.htm

---

[1] http://www.crcstudio.arts.ualberta.ca/streetprint/viewtext.php?s=browse&tid=116&route=bytitle.php&start=36
[2] http://www.crcstudio.arts.ualberta.ca/streetprint/viewtext.php?s=browse&tid=115&route=bytitle.php&start=72
[3] http://www.peerfear.org/cnn/



These Weapons of Mass Destruction cannot be displayed

The weapons you are looking for are currently unavailable. The country might be experiencing technical difficulties, or you may need to adjust your weapons inspectors mandate.

Please try the following:

- Click the Regime change button, or try again later.

Figure 1-2, from http://www.coxar.pwp.blueyonder.co.uk/

This initial foray into the history of popular print reassured me that a public digital collection of centuries-old British street literature could indeed preserve (and perhaps even re-emphasize) the inclinations of the original texts. I could justify the fit, having found similarities in both mediums' populist impetuses, modes of transmission and circulation, and even recurring themes and genres.

Around this time, two other aspects of the collection were emerging as further signs that a coupling of street literature with digital technology would prove fruitful. The first of these was a lack of copyright restrictions upon the texts in Kelly's collection. The importance of this factor to the project can hardly be overstated. While enormous legal and ideological debates are currently raging on the subject of copyright and the internet—and I must admit a certain sympathy toward the viewpoints being advanced by the likes of Lawrence Lessig, Cory Doctorow, the EFF, and others[4]—Streetprint was clearly obliged to respect the University of Alberta's institutional guidelines (and Canadian copyright law) on all intellectual property matters. Since Kelly and I had long since decided that our website would be completely free and accessible to the entire online public—in fact, we felt this to be an essential part of solving Kelly's research and pedagogical problems while remaining true to the popular spirit of the material—we were extremely fortunate to be working with a collection which (by virtue of its age, provenance, and "sub-literary" status) had long since been out of copyright. Not coincidentally, the relative absence (or at least unenforceability) of copyright regulation in the late eighteenth- and early nineteenth-century Britain meant that the vast majority of texts in Kelly's collection were in fact, by today's standards, "pirated." Given the internet's reputation as a hotbed of copyright infringement, it was tempting to view the entire issue of ownership as yet another sign that this collection of street literature was meant to go digital. Two years later, of course, Streetprint has blossomed into the Streetprint Engine and copyright concerns—along with questions of licensing, compensation to copyright holders, and models for restricting access—are square on our radar. At the time, however, our freedom from the restrictions of copyright allowed us to develop the initial Streetprint database much more freely and rapidly than would have otherwise been possible.

---

[4] To learn more, good starting points include Lawrence Lessig's homepage (http://lessig.org/) or Cory Doctorow's recent talk on Digital Rights Management delivered to the Microsoft Research Group (http://www.craphound.com/msftdrm.txt).

A final aspect of Kelly's street literature collection which worked to our advantage was not a literary nor a legal consideration, but simply a physical one: the majority of the collection's texts were small enough to fit comfortably on a consumer-grade, 8.5 x 11-inch scanner bed. This allowed us to obtain incredibly sharp, vivid images of the collection's texts with relative (and affordable) ease, cementing our decision to structure Streetprint's database around an image-based representation of each item. We were consistently impressed at how well a basic flatbed scanner—in our case, a $200 Epson Perfection 1200—conveyed the paper texture, colour, and other visual elements of each text. In June 2002, after Chris Govias and I presented a set of sample images to Gary Kelly, it was apparent that all the necessary pieces were in place to begin the project in earnest. It was time to start coding.

Code Before You Think (Too Much)

Choice of programming language—much like the choice between competing operating systems, hardware platforms, text editors, and first-person shooters—is debated amongst computing geeks with near-religious intensity. Largely due to my previous experience and the fact that the CRC Studio's new web server came pre-installed with the language, I decided to code the initial version of Streetprint in PHP[5], using MySQL[6] for the database. Technically, PHP is actually an interpreted HTML scripting language as opposed to a full-blown compiled programming language, although this distinction has become less meaningful with time. Designed specifically for the world wide web, PHP's flexibility is perhaps the language's best asset. Over the past several years, PHP has grown along with Streetprint, allowing us to migrate to an object-oriented design model and add new standards-based features without having to switch platforms or languages. PHP borrows the majority of its syntax from C and Java, well-respected programming languages used frequently in my undergraduate Computing Science courses. By 2002 it had also become a de-facto standard on many academic and commercial web servers, which ensured a potentially wide user base once we transformed Streetprint into the Streetprint Engine and made it available for download. Finally, both PHP and MySQL are free and open source; I'll discuss the significance of using open source software at length in my next chapter.

---

[5] http://www.php.net

[6] http://www.mysql.com

In describing the Institute for Advanced Technology in the Humanities' seminal 1990s work on the Rosetti Archive at the University of Virginia, McGann explains that

> a major challenge for the institute and its fellows was to pursue long-term, large-scale humanities computing research projects with an almost ascetic rejection of the surface effects and short-term gains offered by proprietary software and proprietary data standards...this pursuit reflected a dedication to portability and the abstraction that enables it—even if it also entailed doing without good tools for creating or disseminating the scholarly work in the short run. (*radiant textuality* 10-11)

By 2002, thanks in part to pioneering, standards-making work by contributors to open source software such as PHP and by academics such as McGann, I felt confident that it was possible for Streetprint to avoid this decision between short- and long-term gain altogether. Rather than pursue the more traditional academic computing model of continuing my research for another year or two, working in splendid isolation to craft the most perfect and complete architecture I could imagine, and then finally unleashing it upon an unsuspecting public, I felt that the clear and present need for something like Streetprint was simply too great. I hoped that by borrowing models of rapid prototyping and deployment from the open source software world—*release early, release often*, the saying went—we could have our cake and eat it too. I wanted something that *worked,* and soon. Once I had something working, I reasoned, it would be that much easier to see how it could be made to work better.

By July 2002 I had crafted the very first Streetprint database specification, a document which specified exactly which tables and fields the database would contain. (In a relational database, each "table" represents a discrete concept, such as a collection item, an image, or an association between the two. Every table has fields whose organization determine what sort of information can be collected in that particular table's records.) Although Streetprint's website was going to be image-oriented, I reasoned that it still made the most sense for the underlying database to be organized around the 'texts' table. Every item in the collection would be represented by one record in this table. My initial design for the texts table was exceedingly simple, consisting only of ten fields:

```
mysql> describe texts;
+-----------+----------+------+-----+---------+----------------+
| Field     | Type     | Null | Key | Default | Extra          |
+-----------+----------+------+-----+---------+----------------+
| id        | int(11)  |      | PRI | NULL    | auto_increment |
| title     | text     | YES  |     | NULL    |                |
| author    | tinytext | YES  |     | NULL    |                |
| publisher | tinytext | YES  |     | NULL    |                |
| city      | tinytext | YES  |     | NULL    |                |
| doctype   | int(11)  | YES  |     | NULL    |                |
| category  | tinytext | YES  |     | NULL    |                |
| date_num  | int(11)  | YES  |     | NULL    |                |
| date_text | tinytext | YES  |     | NULL    |                |
| notes     | text     | YES  |     | NULL    |                |
+-----------+----------+------+-----+---------+----------------+
10 rows in set (0.04 sec)
```

Figure 1-3

This initial table was intended solely for Kelly's street literature collection, capturing what I felt at the time were the most salient characteristics of each collection item, elements which might be useful for browsing, searching, and sorting. (All the images associated with this collection item are stored in their own separate table.) There are fields for Titles, Authors (though the majority were anonymous), Publishers, the City where the item was printed, and a catch-all Notes field. Two other fields were for Document Types (doctypes) and Categories, both of which have their own respective tables in the database. The 'document type' of a text refers to its physical *kind*; document types in Kelly's collection included ballad sheets, chapbooks, and broadsides, for example. The

Categories field was intended to function in a similar way, linking to a 'category' table which contained a thematic taxonomy. While looking for a taxonomy I could use for testing, Kelly pointed me toward the subject guide of Indiana University's Lily Library Chapbook Index[7], a repository of material similar in many respects to our own collection. Borrowing the Lily taxonomy, the first version of our 'category' table soon included entries such as "Book of Instruction," "Crimes and Criminals," "Metrical Tales & Other Verse," and so on.

The document type and category fields were central to one of my chief design goals for the first version of Streetprint: encouraging lateral leaps and more exploratory, free-form browsing of the collection. If you were looking at a ballad sheet, I reasoned, it would be nice to be one click away from a list of all the other ballad sheets. Likewise, the ability to associate multiple categories with a single text meant that if the ballad sheet belonged to the "Dramatic" and "Occult" categories, you could click on the name of the latter category and instantly be browsing only the Occult-related texts in the collection, whether they were ballad sheets or broadsides or chapbooks. Although these concepts are exceedingly simple, I had been consistently surprised at how often other online collections forced me to specify, in advance, exactly what I was looking for. Even if I could find an item, I was often "stuck" within my limited search results, without many options for exploring in and around a specific text; it was always back to the search box for another specific query. I hoped that by creating document types and categories, among other features, Streetprint could provide an architecture which encouraged much more exploratory browsing. Moreover, since I had created separate database tables for both concepts (as opposed to simply supplying a fixed list of doctypes or categories in advance), I was able to build Kelly a series of tools allowing him to add, edit, and delete categories and document types, giving him complete control over this aspect of his website's browsing taxonomy.

---

[7] http://www.indiana.edu/~liblilly/lforms/chapbook.html

The final two fields in the initial version of the text table are two date-related fields, "date_num" (a field which only accepts whole numbers) and "date_text" (a text field). These fields represented my first attempt at solving a deceptively simple problem: how could the database store necessarily-imprecise dates in such a way that both humans and the computer could use and understand them? The complicating factor was the nature of the collection's texts. A small minority of the texts were printed with exact dates ("January 17, 1805") or years ("1810"), but determining dates for the majority of the collection involved guesswork ("1790?") or using textual clues to determine date ranges ("ca. 1820-1830"). All these different ways of reporting a text's date make perfect sense to a person, but will only confuse a computer which, when confronted with many different date-writing conventions and asked to put its records in chronological order, would be unable (without prohibitive amounts of coaching) to determine that "ca. 1796-98?" should come before "April 20, 1840." My initial response to the problem was that the human-readable dates should take precedence, and my solution to the problem reflects that impulse. I created the date_text field as the "primary" date field, into which dates such as "1810-20?" could be entered. The second field, date_num, would be used solely for ordering purposes, and its value would never be displayed to a website visitor. The site editor, then, could place "1810-20?" in the date_text field and "1815" in the date_num field, ensuring that a computerized chronological ordering of the texts would be possible, and appear reasonable, to a website visitor. At the time, I felt that this solution was a clever response to an aspect of the physical collection which refused to submit to the precise, ordered world of the machine, an acceptable compromise between text and tech. As it turned out, however, the imminent expansion of the project's scope to include generalized collections—the "a lot of people could use this" eureka moment which came only weeks later and added the word *Engine* to our name—would expose the weaknesses of this particular compromise.

Programming Politics

I have just documented the initial thought process behind the design of an extremely small (and primitive) Streetprint database table. I chose this particular table as one example, one construct which can hopefully stand in for many. The table in question has been expanded and revised more than a half-dozen times since the initial version illustrated in Figure 1-3, and even in its embryonic form it made up a relatively small part of the first version of Streetprint. It still took several pages, however, to describe only some (not all!) of the decisions I made and the compromises I reached while developing this single component of Streetprint. The strictures of current digital technology make this sort of decision-making an inescapable, essential aspect of developing digital tools. Furthermore, these decisions are rarely just theoretical or abstract; in my simple example, the choices I made directly impacted the ways in which Streetprint users and editors could understand and navigate through the popular print collection. In short, I was restricting—or more generously, suggesting—what could be *known* about the complexities of each text in the digital archive.

In his book *Code,* Laurence Lessig's central argument is that "code is architecture." Lessig also credits EFF co-founder Mitch Kapor for the maxim "architecture is politics" ("The Code is the Law" 1). I want to make this connection explicit: ***programming is inherently political.*** Regardless of whether a programmer's preconceived design goals are the driving force behind decisions (my dream of 'exploratory architectures') or these decisions are made in direct response to emergent technological limitations (the 'date' compromise), these choices are never impartial nor automatic; they reflect the very human politics of their digital authors. Although this observation carries no value judgement, failure to acknowledge the political and human authorship inherent in software development can often result in frustrated users who become confused or offended when their software's logic—underscored by (normalized) politics masquerading as machinic neutrality—fails to overlap with their own. Cultural critic Ellen Roses expresses this very frustration with some university-provided course management software, noting that from her point of view,

> ...the [software's] structure itself insidiously compels faculty to accommodate their instructional practices and decision-marking to the imbedded aptitudes of the system...I found myself succumbing to the pressure to fit my course content and instructional strategies into predetermined formats which, premised as they are upon the assumption that an instructor's ultimate goal is to deliver information efficiently, accorded with neither my content nor my convictions. (60-61)

As I suggested earlier, the political impetus of software need not appear "insidious," especially if software providers communicate effectively with their user base. Working in the CRC Studio on the first version of Streetprint, I had the distinct advantage of frequently being in the same room as Gary Kelly, our designer Chris Govias, and many other studio researchers who subsequently became Streetprint users. Our decision to turn the Streetprint Engine into open source software (touched upon later in this chapter as well as at length in

Chapter 2) was another step toward acknowledging our own software's politics and authorship. By posting source code and creating discussion forums, it became much easier for others to examine and critique our code and the values embodied therein. Finally, this chapter's many remaining case studies aim to shed further light upon the politics of programming, both in the specific and the general. Version 2.0 of the Streetprint Engine was largely defined by an architectural transition towards object-oriented programming (OOP), a development model whose central tenets necessarily complicate the concept of individual software authorship. The development of version 2.1's "basic search" feature also provides an interesting glimpse into the world of algorithmic intentionality and the balance of control between software developer and user.

A Lot of People Could Use This Thing

In July 2002, roughly halfway through the development of the first version of Streetprint, Gary Kelly's casual seven-word observation dramatically changed the project's scope. He was right, of course; why build a database, a website, and an entire set of user-friendly content-management tools only to use them for a single collection? A lot of people *could* use something like Streetprint, we reasoned, both within the domain of popular print and beyond. We had grown attached to the populist connotations of the Streetprint name, and so we decided to rechristen our software only slightly, emphasizing its new ability to power many different collections by calling the *Streetprint Engine*. Gary Kelly's original archive of street literature, whose digital version we had intended to call *Streetprint: The Edmonton Collection*, became *Revolution and Romanticism*[8]. It went online in September 2002, becoming the first website powered by the Streetprint Engine, version 1.5[9].

---

[8] http://www.crcstudio.arts.ualberta.ca/streetprint/

Our new mandate also presented challenges far more daunting than a nomenclature adjustment. One of these challenges was a visual design problem. For over a month, Chris Govias had been busy working on the user interface for the collection's website, as well as designing a beautiful logotype and four illustrations in the style of the many woodcuts and engravings which characterized the collection's texts. Each illustration corresponded to an element in the simple navigation bar Govias envisioned would sit at the top of every page on the site: one graphic each would link to HOME (the website's front page), BROWSE (ways to browse the collection), SEARCH (enter your own search terms), and ABOUT (about the collection and contact information). This navigational structure proved simple but effective, and while the illustrations were recognizable enough to instantly communicate their meaning (the sketched outline of a house, the open pages of a book), they also conveyed an intimate understanding of the visual cues and bibliographical coding present in the collection's texts. Combined with an appropriate background colour and careful typeface selection, the effect was striking. The collection's many digital images felt like part of an organic whole rather than outmoded relics of the past, jolted uncomfortably into a digital future.

The problem, of course, was that Govias' design now appeared *too* successful; our expanded mandate seemed to suggest the need for a visual design which could suit *any* given collection. The current design's aesthetic was so clearly wedded to *Revolution & Romanticism* that applying it to a different website was unthinkable. A parallel with an earlier research problem began to emerge, posing the question: if we were unwilling to strip our collection's texts of their unique visual elements (and had decided to give precedence to image-based representation over encoding of linguistic content), why would we consider doing the same to our website? The design of a website may be considered window-dressing by some, but—as with our street literature—we firmly believed that these visual elements were every bit as important to a user's (or reader's) experience as the site's "content."

---

[9] Since we were already using 1.0 as our version number in the pre-Engine development era, it was decided that the first version of the Streetprint Engine would reflect this heritage, and we would call it version 1.5.

The solution came when we realized that in creating our first Streetprint site, we had actually solved two separate design problems: one of information design (how many sections the website should have, the placement of elements on the page) and the other of visual design (the woodcut illustrations, the background colours and textures). The former would likely lend itself well to almost any online collection, whereas the latter was an extension of a specific collection's visual aesthetic. Once this idea had settled in, I realized that we could borrow a concept from the desktop computer world called 'skinning.' The metaphor is apt; essentially, a 'skin' is a series of graphics (such as a logo, woodcut illustrations, etc.) which are then placed upon a 'skeleton'—in this case, the information design and layout of a Streetprint website. This concept becomes most useful, of course, when you implement it in such a way that you can easily 'switch skins' without needing to adjust the underlying skeleton. PHP made it extremely easy to implement skins in Streetprint. By replacing every instance of an image ("display revolution_home.gif here") with a small function that first read a small skin configuration file ("find out what the 'home' image is for this skin and then display it here"), it was suddenly possible to give a Streetprint site an entirely new look simply by dropping some new graphics into the appropriate directory.

In the web design world, 'skinning' is simply another way to say 'separating form from content,' something of a holy grail since the early-1990s invention of HTML and the World Wide Web. The past two years have seen a major—and largely successful—grassroots campaign for the increased adoption of existing web standards, with a particular emphasis on XHTML (an XML-compliant form of HTML) for encapsulating content and CSS 2.1 (Cascading Style Sheets) for encapsulating 'form'. Web browser support for these standards has improved so much in the past several years that it is now possible to store the entire visual design and even layout of a website in a single CSS file. Since the CSS specification is a mature, ratified standard[10] which should last well into the future and ensure accessibility, upcoming versions of the Streetprint Engine are transitioning from our proprietary skins implementation (using PHP) to the standards-based one (using CSS). Nonetheless, Streetprint skins are an excellent example of our ***release early, release often*** mantra in action. In 2002, support for CSS-based layouts was still unreliable, particularly in older web browsers. Unwilling to compromise on the visual design of our websites, we quickly deployed our own solution instead. Although the migration of existing Streetprint sites to this new standards-based system will require some extra work, the benefits we've realized over the past two years certainly justify this effort. Nearly a dozen Streetprint sites have enjoyed the benefits of having their own unique skins which made integral contributions to the visual appeal, integrity, and success of their digital collections.

In order to determine if our newly-conceived Streetprint Engine was truly viable, however, we needed a second Streetprint site. This meant we also needed to find a second collection to digitize. With our intended September launch date rapidly approaching—along with a full graduate course load for myself which would leave little time for coding—a creative solution was proposed. Kelly's collection contained several dozen pieces of popular children's literature, the majority of which came from the nineteenth-century side of the archive. We selected several items from this subset of his collection and used them as the foundation for our second Streetprint site: *Kidprint*.

---

[10] http://www.w3.org/TR/CSS21/

Govias set to work designing a skin for the new site, borrowing his illustration concepts from *Revolution and Romanticism* and setting them in a new style reminiscent of children's building blocks. As he designed, I worked to complete the code for Streetprint's collection management tools so that Kelly could easily edit and maintain both our new collections. I also revised our initial database specification to fit our growing needs, and in response to feedback from colleagues who had seen the early 1.0 version of *Streetprint: The Edmonton Collection*. The 'texts' table, for example, gained three new optional fields: pagination (total number of pages, so that a text's size could be conveyed in cases where we did not provide images of every single page), illustrations (to remark on particular visual details), and dimensions (to keep track of an item's physical size). Finally, finishing touches were added in order to improve ease-of-use and to encourage casual web surfers to explore the site. The "featured text" was one such addition. Created in order to increase the visual appeal of a Streetprint site's front page, this feature allowed site editors to select a text from their collection via a pull-down menu. The selected text's primary image (usually the cover or first page) was then displayed prominently on the site's main page, along with its name, author, and a link inviting users to browse the item's complete record and view more images. The featured text could be changed as often as the site editor liked, providing a convenient way to highlight new content or to illustrate the range of material in the collection. In the past, I had often been frustrated with online collections which—from my point of view as a user—would "hide" actual images of collection items deep within their site, often requiring multiple clicks just to get to a point where images could be browsed. I hoped that the featured texts' prominent image on the front page of every Streetprint site would draw users in, promising something our software could deliver: an immediate, tangible, and powerful connection with the content of our digital archive(s).

Another feature added to the Streetprint Engine in August 2002 was a text-based navigation system affectionately dubbed "the breadcrumb trail." Much like in Hansel and Gretel, the breadcrumb trail indicates a user's current position and provides a series of links back to each step on the "route" they took to get to their location within the website. Implementing this relatively-straightforward concept in PHP was much more difficult than expected, mostly due to an earlier decision (for the sake of backwards compatibility, privacy concerns, and a few other reasons) to avoid the use of cookies or other persistent methods of tracking a visitor's movement within the site. I eventually settled on a slightly inelegant (but functional) recursive method for displaying breadcrumb trails, a method still used today in version 2.1. A typical Streetprint breadcrumb trail looks like this, where underlined phrases link to previously-visited pages:

Home > Browse > By Type > Chapbook > Robinson Crusoe > Viewing Images

Though the concept is simple, the "you are here" effect provided by breadcrumb trails dramatically improves a website's usability, and they have become a staple on everything from e-commerce sites to online library catalogues. The addition of a breadcrumb trail to the Streetprint Engine nicely complemented our existing graphical navigation bar and helped cement our ease-of-use prior to our fall launch.

By September 2002, version 1.5 of the Streetprint Engine was complete. Version 1.5 consisted of 5,288 lines of PHP code, of which 2,555 powered the "front end" (the website the public could visit) and 2,733 lines powered the "back end" (collection management tools the site editor used to maintain the archive). These 5,288 lines of code were spread over 89 separate files. The CRC Studio webserver[11] was running two copies of the Streetprint Engine and its associated database; one for *Revolution and Romanticism*, which contained approximately 120 texts at launch, and another for *Kidprint,* which contained six texts (albeit some of substantial length) at launch. We also created a third website, *Streetprint.org*[12], which acted as a hub for both of our Streetprint sites and for information on the Engine itself. Screen captures from our two initial Streetprint Engine 1.5 sites and *Streetprint.org* are available in Appendix A.

In September 2002, shortly after the official launch of *Revolution and Romanticism* and *Kidprint*, I began my graduate coursework in English at the University of Alberta. During the next eight months, we received excellent feedback on our two Streetprint sites. Though the sites admittedly lacked the maturity and feature-depth of more established humanities computing projects, our populist approach, attention to design, and ease-of-use—to say nothing of the compelling texts themselves—all garnered praise from students, researchers, and web surfers. I had little time for coding during this period, but I was becoming increasingly aware that version 1.5 of the Streetprint Engine also had a few major problems. Because the physical texts in *Kidprint* and *Revolution and Romanticism* were so similar, most aspects of Streetprint's architecture remained inflexibly tied to the specificities of our two collections. This rigidity directly conflicted with our long-term vision of the Streetprint Engine as a software system for all kinds of digital collections, and for all kinds of researchers and communities. In the CRC Studio, for example, Gary Kelly was planning a 2003 summer research institute on popular literature in Edmonton. Many of the student researchers were hoping to create a variety of Streetprint sites showcasing their projects. Their research was unusual and exciting—involving material ranging from theatre programs to contemporary rave and electronica music flyers—but I was worried that the current Engine would be unable to scale up to meet their varied demands. More generally, we had another problem: Streetprint was confined to the CRC Studio. How could we make the Streetprint Engine available for the whole world to use? Though Streetprint worked well on our own webserver, it lacked many of the essentials for publicly available software: we had little documentation and no user manual, no automated installer, no licensing model, no cross-platform testing. From a software development point of view, things were equally problematic. Version 1.5's architecture was largely defined by a *lack* of architecture; my code was idiosyncratic, and many things were hardwired where they should have been abstracted, a symptom of the hectic four-month development schedule and the absence of peer review of the code. By May 2003, my coursework was complete and I could finally tackle these problems. It was time for Streetprint 2.0.

---

[11] http://www.crcstudio.arts.ualberta.ca

[12] http://www.streetprint.org

Streetprint: The Next Generation

The development of Streetprint 2.0 took place between May and October of 2003, and was characterized by three distinct phases. The first was a consultation phase with undergraduate students from Gary Kelly's summer research institute on popular literature in Edmonton. These researchers had a wide variety of needs, technical skills, and ideas about how to report their research, and they provided refreshing thoughts on how Streetprint could be used. The second phase involved taking some of their suggestions, combining them with feedback from the previous year and with our own ideas, and then coming up with a comprehensive list of new features and improvements we wanted to build into the next version of the Engine. The third phase was certainly the most challenging: adding a second software developer to the coding team, then implementing all the new features on our wish-list while completely reworking the Engine's architecture to be more in line with the principles of object-oriented programming, all with the final goal of releasing the Streetprint Engine to the public under an open source software license. This section examines each of these phases in turn.

The website which grew out of the 2003 Edmonton Popular Literature Project, *Cityprint: Edmonton*[13], expresses the research program's goal of using digital technology "to capture, conserve, and communicate the fleeting and ephemeral world of popular print, its place in its readers' daily lives, its cultural and social significance" ("About Cityprint" 1). The site goes on to explain that

> each Cityprint project asks a number of questions: What are people reading? Where do they obtain it? Where does it come from? Once we have some answers to those questions, we can go on to others: Why do people read what they do? How important is it in their lives? How is their reading matter valued by themselves and by cultural guides--teachers, publishers, critics, and so on? ("About Cityprint" 1)

Since the Cityprint research program was centered around questions of circulation, communication, the popular and the ephemeral, I was hopeful that the Streetprint Engine could play a productive key role in both the conduct and the reporting of the project's research. As it turned out, more than half of Cityprint's eight projects eventually included a website powered by the Engine; I'll consider two examples here.

---

[13] http://www.crcstudio.arts.ualberta.ca/cityprint/

Garvin Cheung had just convocated from the University of Alberta with a Bachelor of Science degree when he began working in the CRC Studio in May 2003. His degree didn't hint at his passion, however: Edmonton's rave and electronic music scene. His summer research focused primarily on the print materials which circulated within this culture. Cheung's large collection of music print (augmented by donations from friends and collectors in Edmonton as well as other cities) presented a wonderful opportunity for a contemporary digital archive powered by the Streetprint Engine. Not only were the collection's posters and handbills a fundamentally visual medium, but they were "street print" in the truest sense, free printed ephemera placed on bulletin boards, lamp posts, and distributed at nightclubs, material which circulated briefly and then vanished permanently from sight. In an afternoon brainstorming session with Cheung and the CRC Studio's two visual designers (bolstered by the recent addition of designer Hannah Wensel), we even hit upon a perfect name for the archive, one which played upon both musical and digital themes: *Urban Record*[14].

Cheung was a relatively savvy computer user, and so he quickly grasped which elements of Streetprint needed modification in order for *Urban Record* to succeed on its own terms. Not surprisingly, terminology topped his list: Cheung preferred to use the word "print" instead of "text" when referring to items in the collection, and he also wanted to rename a number of Streetprint's fields from the 'texts' table: "creator" instead of author, "venue" instead of publisher, and so on. The flexibility built into Streetprint's document types and categories meant that these two fields could meet Cheung's needs without modification. He planned to use *Urban Record*'s document types for physical distinctions ("Card," "Flyer," "Poster," "Sticker") and categories to distinguish subject matter ("Advertisement," "Club Night," "Festival," "Party/Rave").

---

[14] http://www.crcstudio.arts.ualberta.ca/urbanrecord/

One aspect of Streetprint which didn't bend so gracefully to suit Cheung's collection was the way it handled dates. As previously discussed, Streetprint's implementation of dates was tailored to a collection in which having a specific year was as precise as one could hope to get. Many of Cheung's posters, however, listed exact dates for concert events. If he entered this date ("November 5, 1999") into the date_text field and then the year into the date_num field ("1999"), users of *Urban Record* who viewed his collection in chronological order would be unimpressed; records would appear out of order, since exact dates would be displayed but the sorting mechanism would sort only by year. Moreover, some items in Cheung's collection presented further complications: how should the poster for a three-day festival be catalogued? What about a 2001 club poster advertising a theme night "every Tuesday?" I knew that a long-term solution to this problem would take time and research, so Cheung and I decided that in the meantime, only year values would be entered into both the date_num and date_text fields, ensuring that *Urban Record* wouldn't promise more information than it could deliver.

Over the course of the summer, Cityprint researcher Andrea Hasenbank conceived of a very different idea for Streetprint site. Her research project was an investigation of genre in popular paperback fiction. Hasenbank had assembled descriptions and examples of each genre, as well as a visual taxonomy illustrating some hierarchies and relationships between genres and sub-genres.

From my point of view—set in my ways as I was—Hasenbank then had what was a fairly radical thought: her research had created a "collection" of genres, so why not digitize it using the Streetprint Engine? Once the implications of this idea set in, it became apparent that repurposing Streetprint for a collection of genres (instead of texts) would actually require very few changes to the software's functionality; it again became a matter of tweaking terminology. Beyond obvious changes, such as turning the "featured text" into the "featured genre," most of the text table fields would also need renaming. "Author," "Publisher," "Location," and "Notes" would give way to "Conventions," "Examples," and "Description," for example.

More innovative were Hasenbank's ideas on how to use document types, categories, and images. The "traditional" use of document type was preserved; Hasenbank elected to use "paperback novel" as the site's sole document type, leaving the door open to the future addition of genres from other kinds of texts. Categories, however, would be used for major distinctions in genre, for the headings near the top of her taxonomic tree of genres and sub-genres. The entry for "Time Travel Romance," for example, would belong to the "Romance" category, and the entry for "Dark Fantasy" would belong to both the "Fantasy" and "Speculative Fiction" categories. Despite no previous experience with databases, Hasenbank had found an effective way to represent her taxonomy within the preexisting categories infrastructure provided by the Streetprint Engine.

The Streetprint Engine allows site editors to associate one or more images with any given text; in fact, since its mechanisms for browsing are centered around images, every text *requires* at least one image. In *Revolution and Romanticism* and *Kidprint* we would often digitize every page of longer works, and the resulting images were then added (along with page numbers and optional captions) to the text's record. Hasenbank again envisioned a way to use a preexisting Streetprint feature to her own ends; she would add images of different book covers, all belonging to that particular genre, as multiple "pages" of a single text (really "genre") record. (She would also enter the title and author of the book pictured into the "caption" field for each image.) This clever solution added a valuable visual element to her taxonomy, and proved to be an excellent alternative use of the Streetprint Engine's image management tools. Hasenbank named her site *Text in Transit*[15], explaining that the project "brings together transitory pieces of popular literature: portable, forgettable, ephemeral and tossable…by navigating through the categories presented, we can recognize and come to understand what ordinary people are reading as they move through their daily lives" ("About Text in Transit" 1). Screen captures of both *Urban Record* and *Text in Transit* are available in Appendix A.

By the end of May 2003, the CRC Studio team had finalized a definitive list of features we hoped to include in version 2.0 of the Streetprint Engine. Some of these features simply gave site editors control over elements which had previously been "hard-coded." (I hoped to reap personal benefits from the autonomy these new features would bring, since "hard-coded" usually translated to "ask Matt to change it" in practice.) For example, a new 'artifact name' feature would allow editors to change the word—in both its singular and plural forms—used to refer to items in their collections. The editor of a Streetprint site featuring posters, for example, could change the artifact name "text" to "poster." This word would then automatically be used as needed; the front page of their site would now display a 'featured poster' instead of a 'featured text.'

---

[15] http://www.crcstudio.arts.ualberta.ca/textintransit/

Similar features would let editors modify the text blurbs which appear in various locations around a Streetprint site, making the process as easy (and instant) as typing their text into a box and clicking a 'submit changes' button. A new screen called 'search and browse options' would allow editors to specify which fields were most important to their collection; Streetprint would then structure its website's "Browse" and "Search" pages accordingly. This meant that simpler sites could have simple browsing options, whereas a collection which carefully tracked optional elements such as City, Publisher, Dates, or (in the case of poetry) First Lines could provide all these sorting and searching options to a website visitor.

One problem with digital archives powered by version 1.5 of the Engine was that their strengths lay primarily on the *digital* side of things rather than the *archival* side. Streetprint was originally conceived as a one-person solution to a one-person problem, and so we now needed features to address the expanded communal scope of the project. A crucial first step would be the addition of a 'Users' database table and associated features, allowing multiple site editors to register accounts on a single Streetprint site. Anticipating larger team projects, we also decided to create two classes of user: 'site editors' who could use their site's administrative tools to their fullest extent, and 'data entry users' who could only add items to the collection and edit the records of items they had added[16]. The most important aspect of this new Users table was that it allowed the Engine to keep track of who was creating and modifying its underlying database, preserving a "record history" alongside each collection item. Metadata of this sort is crucial to any digital archive, and its omission from earlier versions of the Engine constituted a critical oversight which we were eager to correct. We planned a new 'publish/unpublish' feature which would also enhance editorial control for larger collections, especially those with more than one editor. When an item is first added to a Streetprint collection, it would be 'unpublished.' This means that it would only be visible to site editors, and not someone browsing the website, until an editor deemed the item (and its related information fields and images) ready for public consumption and clicked the 'publish' button.

---

[16] It is worth noting that Team Streetprint *does* occasionally incorporate features which anticipate future needs, despite our semi-official policy favouring features which respond to existing ones. I mention this here because the CRC Studio has yet to use the 'data entry user' feature. Perhaps our populist tendencies compel us to make everyone an editor…

Some features planned for version 2.0 were wholly new. Since the 2002 launch of *Revolution and Romanticism*, for example, Gary Kelly had been seeking a way to easily integrate the archive's material into his teaching. In response to this request we planned an entire suite of course-related tools, allowing site editors to easily construct any number of course homepages and syllabi. These online syllabi would provide all the basics—course name, description, location, and so on—along with a reading list furnishing direct links to all the collection items selected for the class by the instructor. These links could also be augmented by a list of non-digital course materials as well as file uploads, allowing the instructor to attach PDFs, Powerpoint slides, or any other files to the syllabus. By clicking a checkbox, a site editor could then automatically place a link to this syllabus—such as "English 383 students click here"—on the front page of their Streetprint website.

By far the biggest feature planned for version 2.0 wouldn't be explicitly visible to site editors or website visitors at all. This under-the-hood change, however, was a prerequisite for many of the Streetprint Engine's goals, such as ensuring the project's long-term sustainability and growth, our commitment to standards and open access, our desire to move from a single software developer (myself) to a development team, and our aim for excellence in both the humanistic *and* the technical domains. A culinary analogy best captures the essence of this new feature. "The Pasta Theory of Programming," explains an online dictionary, "is the idea that various programming structures can be likened to the structures of well-known pasta dishes" (SearchVB.com 1). The definition's next statement is, unfortunately, the best analogue for my coding style in version 1.5 of the Streetprint Engine:

> The first and most famous example of the theory is spaghetti code, which illustrates the unfortunate tendency of unstructured procedural programming to result in code with little or no structure, making it difficult to understand and update. (SearchVB.com 1)

A far more desirable option is *ravioli code*, which by comparison is

> made up of small, separate, and loosely coupled objects that can be individually modified without affecting the other components or the structure as a whole (SearchVB.com 1).

Ravioli code is better known as the dominant paradigm in modern software development, object-oriented programming (OOP). Since the day the word *Engine* was added to my project's name, I had known Streetprint would need to migrate to an object-oriented architecture sooner or later. As we finalized the 2.0 features list, Team Streetprint also grew to include a second software developer, Matthew Bouchard. Bouchard's dual background in Computing Science and Creative Writing made him a natural fit for the CRC Studio, and he arrived just in time to lend an invaluable outsider's eye to the difficult process which lay ahead: changing Streetprint's menu to include less spaghetti and more ravioli.

OOP, I Did It Again

The history of object-oriented programming can be traced back more than thirty years. 1967 saw the development of Simula-67, a programming language now credited for the introduction of OOP's key concepts (Dahl et. al par. 1). While people often speak of "OOP Programming Languages"—making statements such as "C++ is an OOP language, whereas C is not"—this attribution is not entirely accurate. In fact, the central tenets of OOP have been recognized for decades as good programming practice, and OOP's design philosophy can be applied (in varying degrees) to nearly any programming language. Specific programming languages which have grown in popularity in the last decade, such as Java and C++, can be said to *support* OOP, however. As Anton Eliëns explains, "a language *supports* a style of programming if it provides facilities that make it convenient (easy, safe and efficient) to use that style" (Eliëns par. 5). Streetprint's language of choice, PHP, has provided "object oriented syntax support" since the release of PHP 3.0 in 1998 (Bakken et al. 1). Originally introduced simply as "syntactic sugar," the popularity of this object oriented syntax surprised even the language's developers, who noted that the ability to write in an object oriented manner "proved to have a much more far-reaching effect on PHP than was originally intended" (Suraski 1). The upcoming 5.0 version of PHP greatly improves the underlying implementation of the language's object syntax, promising to bring it very close to Java's object model.

The principles of object-oriented design revolve primarily around five terms: abstraction, encapsulation, information hiding, inheritance, and polymorphism. *Abstraction* and *encapsulation* are perhaps the easiest terms to understand, as their OOP definitions are similar to the literal connotations of the words themselves. The essence of abstraction is "to extract essential properties while omitting inessential details" (Berard par. 6). The "objects" essential to OOP are *abstractions* of this kind, constructs designed to include only the information necessary to the task at hand. In a program concerned with shapes, for example, there might be a Circle object. While there are a number of ways to describe the size of a circle (such as radius, diameter, and circumference), an ideal Circle object would store only a single value, its radius. Values such as diameter or circumference are unnecessary, since they can be easily calculated from the radius. This single, simple "model" of a circle can then be used (and re-used) every time the program calls for a circle. Once a suitable abstraction is found, it should be *encapsulated.* Encapsulation is simply a recommendation about the logical structure of a program's code. It specifies that each object—that is to say, each abstraction—should be separated logically and semantically from the rest of a program's code. This makes sense from a complexity standpoint: if some detail about a circle needs to be changed, a programmer will know exactly where to find it and won't need to go fishing through the entire code, searching for every mention of a circle. Closely related to abstraction and encapsulation, *information hiding* is perhaps the most important tenet of object-oriented design. Information hiding is remarkably similar to Bruno Latour's conception of the "black box." In *Science in Action*, Latour writes that "the word **black box** is used by cyberneticians whenever a piece of machinery or a set of commands is too complex. In its place they draw a little black box about which they need to know nothing but its input and output" (2-3). Information hiding enforces the black-boxing of code objects. It specifies that the implementation details of every object—of every encapsulated abstraction—should be hidden from the rest of the program. This means that interaction between objects should take place through their *interfaces*, rather than their inner workings. Let's say, for example, that our computer program needed to know the circumference of one of our Circle objects. Prior to OOP, this would be done directly: the program would look up the value of the "radius" variable stored inside the circle in question, and would then do some basic math (2πr) in order to obtain the circumference. What would happen if, for some reason, the implementation of Circle object was changed so that it stored diameters (equal to the radius times two) instead of radiuses? Our other code would either break entirely—because it would no longer be able to find the "radius" variable at all—or else it would use the diameter variable in place of the radius, producing a number twice as large as before. (Even in this simple example, it is easy to see how quickly program code can begin to resemble spaghetti, where every piece is confusingly intertwined.)

The use of information hiding completely changes this scenario. With information hiding, the rest of the program can't see "inside" the Circle, so it has no way of knowing if it stores radiuses or diameters. Instead, the rest of the program must interact with the Circle through its declared "interface"; in this case, the Circle's interface would likely include a command called "getCircumference." When the program uses this command, *the Circle itself*[17] makes the necessary calculations to provide a number for its circumference. The rest of the program doesn't need to worry about the feature's implementation or details about whether Circles store radiuses or diameters. The Circle object is now a black box; only the inputs and outputs matter. The advantage to programmers is obvious: as long as the *interface*—the rules for interacting with a program's black boxes—remains stable, the internal *implementation* of objects can be improved or changed completely without "breaking" the program. Two final elements of object-oriented design, *inheritance* and *polymorphism*, are simply extensions of these foundational concepts. Inheritance specifies that objects can be based upon other objects. We could create a "child" of the Circle object called Red Circle, for example, which would inherit Circle's ability to calculate its own diameter but which could also contain colourful properties that Circle lacks. When broadly applied, inheritance permits the incremental construction of incredibly complex hierarchies, nesting multiple black boxes like a series of Russian dolls. Finally, *polymorphism* refers to a program's ability to "accept" multiple object types which share a consistent interface. In our example program, a piece of code might ask an object for its circumference, not caring whether the object was a Red Circle or simply a Circle. Polymorphism greatly reduces the amount of code a program needs to handle a wide variety of situations, since it permits a program to deal with a variety of related objects in a single, consistent manner.

---

[17] That is to say, programming code located inside the Circle object handles the calculation. Assigning subjectivity to objects through phrases like this is common practice in object oriented discourse; this language perhaps reinforces the problems of intentionality I discuss later in this section.

A central objective of object-oriented programming is the promotion of code re-use. One frustration which characterized the first several decades of computer programming was that even the most mundane tasks had to be coded by hand; every new program necessitated the re-invention of the wheel. It is not surprising then that many of the tenets of OOP resemble Latour's black box, a conception which, by Latour's account, is essential to the incremental development of technoscience. The history of OOP can perhaps best be understood as the creation of a standardized infrastructure for the black-boxing of software development. (In fact, a form of object-oriented programming composition called *aggregation* is directly concerned with the combination of multiple code objects into systems which can then be considered as a whole.) What's more, computing scientist Wesley Phoa is able to describe this modular nature of OOP in terms which would make a post-structuralist blush:

> Software modules do not have frozen roles, but are designed to be reusable in different contexts, and possibly to play different roles in different

> contexts. Modules are always underdetermined and extensible, and can always be made more specific in function when necessary. *But,* modules still have well-defined, stable descriptions within a *given* context. We could have gotten all of these ideas from Derrida, if we had not already been using them for years in object-oriented programming. (16)

Regardless of whether these ideas originated with Derrida or the computing scientists, it is certainly true that that the five key tenets of OOP—abstraction, encapsulation, information hiding, inheritance, and polymorphism—have been instrumental to the success of the interconnected, complex software systems which characterize our modern world.

Earlier in this chapter, I argued that programming is always a political act, that architectural and algorithmic decisions are constantly being made by software developers. Critics such as Ullman and Rose contend that representations of computers and software in the popular imagination largely erase these traces of human intentionality and authorship. (Ullman darkly jokes that this is perhaps for the best; after recounting the story of one weekend-long marathon coding session, she remarks that "if people really knew how software got written, I'm not sure they'd give their money to a bank or get on an airplane ever again" (2).) If software which adheres to the principles of OOP indeed resembles a series of black boxes, then we must also contend with Latour's assessment that the black box is hardly an "unproblematic object" (131). It often requires prohibitive amounts of time and energy to "reopen" black boxes, to question their underlying assumptions and the logic which dictates their transformation of input into output. This chapter has hopefully opened some of the Streetprint Engine's black boxes, making them available for closer scrutiny. One of this chapter's final sections outlines my anxiety over the development of one of the Engine's more complicated black boxes, version 2.1's "basic search" feature. In general, however, it remains true that the human intentionality of a computer program becomes much more difficult to uncover when a software developer's political expression is only partly visible, most of it becoming hidden over time inside what Latour calls "tightly sealed" (80) and "darker and darker" (253) black boxes.

When computer programs exhibit puzzling, unanticipated, or even "insidious" behaviour, it can often be attributed to the failure of software developers to interrogate the contents of the black boxes they are deploying within their own code. This negative aspect of object oriented code re-use is humorously demonstrated in a real-life (though embellished) anecdote which circulated widely on the internet in recent years. While coding a helicopter flight simulation program for the Australian military, the story goes, some programmers made a last-minute decision to add some kangaroos to the simulation. Their logic for this decision was that disturbed herds of animals could alert enemy ground forces to an aircraft's position. As the anecdote explains,

> being efficient programmers, they just re-appropriated some code originally used to model infantry detachments reactions under the same stimuli, changed the mapped icon from a soldier to a kangaroo, and increased the figures' speed of movement.
>
> Eager to demonstrate their flying skills for some visiting American pilots, the hotshot Aussies "buzzed" the virtual kangaroos in low flight during a simulation. The kangaroos scattered, as predicted, and the Americans nodded appreciatively . . . and then did a double-take as the kangaroos reappeared from behind a hill and launched a barrage of stinger missiles at the hapless helicopter. (Apparently the programmers had forgotten to remove "that" part of the infantry coding). (Snopes.com par. 6-7)

While the anecdote is a humorous one, it does illustrate one of the dangers of wholesale code re-use and software black-boxing. It would be pessimistic, however, to suggest that the erasure imposed by ever-darkening boxes is always negative or undesirable. For example, the growth of the internet has led to an explosion of collaborative object oriented programming, inspiring many software providers to publicly publish the APIs (application program interfaces) for their code. As more and more software systems (including the Streetprint Engine) move to standardized data formats with open APIs based upon OOP principles—creating more "public" black boxes—technologies are being combined and connected in exciting ways which their original programmers might never have predicted. In 2002, for example, the search engine company Google held a programming contest to celebrate their third anniversary. The winner was a programmer who added geographic capabilities to Google solely through the use of public APIs:

> Daniel [Egnor] converted street addresses found within a large corpus of documents to latitude-longitude-based coordinates using the freely available TIGER and FIPS data sources, and built a two-dimensional index of these coordinates. Daniel's system provides an interface that allows the user to augment a keyword search with the ability to restrict matches to within a certain radius of a specified address (useful for queries that are difficult to answer using just keyword searching, such as "find me all bookstores near my house"). ("Google Programming Contest," par. 2)

The fact that a single unpaid programmer working in his spare time could create such a system is nothing short of remarkable. It attests to the power of black-boxing, open standards, and the collective human intelligence embodied in the multiple layers of software which now mediate nearly every aspect of our 21st-century lives.

Evidence of this power—along with a desire to honour the populist roots of the Streetprint project—also inspired a decision to release version 2.0 of the Streetprint Engine's code as open source software. Open source software is the topic of my next chapter, but it is worth noting here that the decision to go open source nicely complemented our transition to an object oriented architecture. Although we were now creating our own black boxes, we were (for the first time) simultaneously placing them in the public domain. We hoped that this move would shed light on our newly created architectures, allowing Streetprint to reap the benefits of both OOP and open standards.

Streetprint 2.0

After four months of development and an initial beta release in September 2003, version 2.0 of the Streetprint Engine was released for public download under the terms of the GNU Public License[18] (GPL) on October 28, 2003. Version 2.0's codebase was slightly more than twice the size of its predecessor, consisting of 13,796 lines of PHP code. Our object code comprised 2,926 of these lines, another 2,941 powered the "front end" (the website the public could visit), and the remaining 7,929 lines of code powered our newly-expanded "back end" with its many new features. These 13,796 lines of code were spread over 123 separate files. Along with the information and downloads provided at our own *Streetprint.org* website, we also registered the Streetprint Engine project with *Sourceforge*[19], the world's largest open source development website. Sourceforge.net provides free hosting of open source projects, along with a suite of tools to help manage collaborative development, file releases, publicity, and other considerations. Matthew Bouchard and I had been using Sourceforge's CVS[20] servers since July, an excellent service which greatly simplified our newly-collaborative development workflow. Registering with Sourceforge also helped us announce our presence in the burgeoning world of open source software, improving our online profile.

---

[18] http://www.gnu.org/copyleft/gpl.html

[19] http://sourceforge.net/projects/streetprint/

[20] CVS stands for "Concurrent Versions System," software which allows multiple software developers to easily keep track of changes to source code files and, as the name suggests, work concurrently.

The vast majority of our planned features and upgrades made it into version 2.0 of the Engine. We now had course and syllabus features, an automated installer, a new "default skin" for freshly-installed Streetprint sites, many site customization options for site editors, a completely redesigned interface for our back-end administrative tools, searchable fulltext records, and much more. Our new architecture, while not a perfect vision of pure OOP, was a definite improvement over the spaghetti code of version 1.5. While we still lacked a comprehensive user manual—I had been too busy writing code instead of writing documentation—version 2.0 included a Read Me file which explained the installation process, offered a few pointers for getting started with your own Streetprint site, and encouraged users to use the new Streetprint.org Forums[21] to ask questions and to give us feedback. Chris Govias' new visual design for the Engine's administrative tools was so clean and easy-to-use that we hoped the majority of users would hardly need written documentation at all. By November 2003 we had also upgraded all five of the CRC Studio's existing Streetprint sites so that they were powered by version 2.0 of the Engine. Now that the Streetprint Engine was fully meeting our own needs—for the time being, at least—the general consensus in the studio was that our software was finally ready for prime-time. Streetprint was ready to meet the world.

The CRC Studio was still in its infancy in 2003, and we were gradually learning how to function effectively as a digital research unit which aspired to connect with communities both within and beyond the scholastic sphere. Since Streetprint was originally conceived as a solution to a literary research problem in the field of popular print, it made sense at the time to begin the promotion of the Streetprint Engine within the academic community. We held a launch party at the University of Alberta to coincide with the software's official release, and Gary Kelly and I gave presentations on Streetprint at every opportunity, most prominently in December 2003 at the annual convention of the Modern Language Association in San Diego. Reactions to Streetprint at these conferences were illuminating, throwing the project's strengths, weaknesses, and relationship to more "traditional" humanities computing projects into sharp relief.

---

[21] http://www.streetprint.org/forums/

Responses to what I felt were Streetprint's strongest attributes—such as our integral attention to visual design, or the fact that our collection(s) and the software itself were completely free to use—were overwhelmingly positive. And to my slight surprise, many academics in the humanities had at least some understanding of open source software, and even those who didn't were receptive to the concept and avoided questions such as "how do you plan to make money?" (This was perhaps due to their familiarity with public funding, granting agencies, and the many other alternative forms of economic viability unique to universities.) One question which arose time and time again, however, was the issue of Streetprint's (relative lack of) metadata support and, by association, support for fulltext encoding methods. Because Streetprint broke the mold with its focus on image-based representation (over textual markup) and a home-spun list of data fields (over standardized field names corresponding to various schema for digital text encoding), we were constantly being interrogated with regard to Streetprint's "academic" value. While my answer to these questions was an honest one—usually along the lines of "metadata is on our radar and support will be built into a future version, but we've accomplished a great deal without it" with the occasional addition of "we're open source, so we'd love your help in coding this feature!"—it never seemed to satisfy our critics. The underlying implication of these questions, it seemed to me, was that Streetprint lacked academic rigour and was therefore not a "serious" research project. Much to the frustration of Kelly and myself, this implication also seemed tied to the nature of the popular print which comprised the bulk of our collection(s); it became clear that some of our colleagues would much prefer projects along the lines of a "rigorous" encoding of the complete works of Shakespeare into carefully architected XML. In this world view, Streetprint was not-very-serious software for collections of not-very-serious literature. This line of reasoning was nothing new to Kelly, who has practically made a career of elevating subjugated texts and voices into the realm of "serious" research, and he largely shrugged off these concerns. Despite what the text-encoding community thought of Streetprint, Kelly's view was that the software had already paid off handsomely in the CRC Studio, and would sooner or later be discovered by others who could use it. (And, unlike those XML works of Shakespeare, our collections were available online to anyone in the world, now.) I was not able to ignore these criticisms as easily. For one thing, despite the politics often couched within their line of questioning, I realized that the eventual addition of standards-based metadata support to Streetprint was absolutely crucial. No database should be an island, and proper metadata would allow the contents of Streetprint collections to become "black boxes" within the larger global collections (and connections) made possible by the internet. I also was reminded of the politics which underscored Streetprint's philosophy of rapid public prototyping and development. Couldn't we have our cake and eat it too? I wondered to myself:

who exactly is the audience for this software? Could the Streetprint Engine find a way to please the populists *and* the academics?

As our efforts to raise awareness of Streetprint within the academic community continued, I realized that the audience for our software now consisted of two different kinds of "users." The first group was the larger of the two: web surfers who could use Streetprint by browsing and searching the various Streetprint collections (such as *Revolution and Romanticism*) to which we linked on *Streetprint.org.* These were the users Gary Kelly had in mind when Streetprint was originally conceived, the students and researchers who couldn't handle the physical texts but could now get at them digitally. Here, our publicity efforts were visibly succeeding. The number of "hits" on our websites was rising steadily, bolstered by links to *Streetprint.org* and our collections from academic and personal weblogs, eighteenth-century resource pages, book design sites, community folklore portals, Google searches, and more. Another group of potential users, necessarily much smaller, had come into being once we released version 2.0 of the Engine. This was the group that we generally had in mind now—as software developers—when we imagined our "users." This group consisted of Streetprint site editors, people who would download the actual Engine, install it on their web server, and use it to power unique new collections. Now that we were making the Engine freely available, it was our hope that this group of users creating their own Streetprint sites would grow at a rate proportional to our more casual web-surfing audience. My personal dream was that, overnight, a constellation of Streetprint servers would spring up around the globe, powering digital collections of every shape and size. Since we were open source, bug reports and improvements would pour in from every continent, and Streetprint would take on a whole new life of its own. Birthed in the CRC

Studio, our software was now released into the wilds of cyberspace to live free and prosper.

Not surprisingly, Streetprint's progress in this arena didn't quite live up to my idealistic reverie. In fact, although the CRC Studio now hosts roughly a dozen separate Streetprint collections, the process of external third-parties using the software to their own ends is only beginning now, nearly a year after the initial release of version 2.0. Many reasons can be given for the relatively slow uptake of the Streetprint Engine within the broader community: we had done virtually no marketing save a few conference presentations; it was possible that the "do-it-yourself" energy and geekery of the open source world had yet to spill over into the humanities; Streetprint was easy-to-use, but perhaps it could be easier. Emerging from these thoughts were two important questions which shaped the next year of the project's development: How could we make the Streetprint Engine even better? And would communities outside academia be interested in it too?

Not Invented Here, But You'll Like It: Streetprint 2.1

After presenting at another conference where Streetprint was very well-received but where few colleagues seemed interested in starting sites of their own, I was reminded of a term from the computer industry: "Not Invented Here" syndrome. *Wikipedia*'s definition explains that

> Not Invented Here (or NIH) refers to the problem when people in companies continue to ignore existing solutions to problems because it was not created in-house. It is endemic to the computer industry (but by no means limited to it). In many cases NIH occurs as a result of simple ignorance, as many companies simply never do the research to know if a solution already exists. But equally common are deliberate cases where the

> engineering staff rejects a solution, typically because they believe they can do better. ("Not Invented Here" definition, 1)

NIH is every bit as endemic to the academic world as it is to the corporate one. This is particularly true in the field of Humanities Computing, where nearly every conference paper tells the story of how the presenter developed their own system or software. This is understandable, since "I found this free software online and then I used it" rarely constitutes a substantial academic declaration. It would also be hypocritical to disparage NIH completely, since its motivating impulses contributed to the creation of Streetprint Engine itself; no one was providing an exact solution to our problem, and so we coded one ourselves. What we began to realize, however, was that promoting greater use of the Engine within the academic community would be difficult unless the "barriers to entry" were extremely low. Streetprint had to be *so* good, and *so* easy to install, use, and customize that any quibbles over smaller aspects of the software would be dramatically eclipsed by its strengths. Potential Streetprint users had to be given an offer they couldn't refuse.

I felt that a good place to begin was by writing something the Streetprint Engine had been lacking since the beginning: a comprehensive user manual. Version 2.0 of the Engine came with a small Read Me file, but I suspected that novice computer users (though expert researchers, curators, and collectors) might be deterred by our lack of documentation and by the fact that the first step in using Streetprint—the installation process—was among the software's trickiest aspects. Open source software is also not known for the quality of its documentation, and so I hoped that a professional, well-designed user manual could help set our project apart. As I began writing what became the Streetprint User Manual, I was presented with a challenge: how could I effectively communicate the process of using our software to set up and maintain a digital collection? Many of a Streetprint site's key building blocks—such as the creation of appropriate document types and categories—worked best when tailored to the specificities of the collection at hand. What examples should I use? While brainstorming with our lead designer, Chris Govias, I hit upon a novel solution: we could *invent* a fictional archive, and then gradually build up a Streetprint site showcasing this material over the course of the manual's 70 pages. In an attempt to counteract what we feared would be the inevitably dry (read: boring) tone of the manual, we decided that our fictional collection would center around a collection of 1940s and 1950s Hollywood B-movie posters, a visual aesthetic near to Govias' heart. I also kept myself amused by weaving hints of plot and back-story into the letters and telegrams in our "collection." Thus was born the tale of Old Man Grinnel, an eccentric Californian movie producer who fled to Canada during the McCarthy era—his collection of Hollywood memorabilia in tow—and settled in the sleepy town of Milk River, Alberta. The complete Streetprint User Manual can be found in Appendix B. Its accompanying website, *Old Man Grinnel and the Milk River Archive,* is also online at www.streetprint.org/oldmangrinnel.

While I was writing Streetprint's user manual, Matthew Bouchard and I were also working on improving the Engine itself. At conferences, Kelly and I had frequently emphasized the need for community tools (which fostered user feedback and interaction) built into the architecture of digital humanities projects. Too often, we felt, worthy projects in humanities computing limited the scope of their audience, failing to take advantage of the internet's global reach or to push beyond their field's pre-digital research boundaries. This emphasis betrayed an embarrassing omission, however. While we had the

Streetprint.org Forums, Streetprint sites themselves remained one-way channels of communication. Aside from sending an email to the site editor, there was no organic way to leave feedback or discuss a particular item on a site. As a first step towards remedying this situation, we decided to add a "user comments" feature. This feature allowed site visitors to attach a comment to any collection item, as well as to read existing conversations. Govias also came up with a unique visual legend—a subtle diamond symbol displayed next to an item's thumbnail image, with a black diamond indicating a recent comment and a white diamond indicating an older comment—which nicely integrated the commenting system into the Streetprint searching and browsing experience as a whole. While basic, the user comment feature marked the beginning of a fundamental shift for the Streetprint Engine, as our digital archives increasingly acknowledged the multiplicity of voices, connections, communities, and communications made possible by the internet—and the architectures we bring to it.

As we searched for additional ways to improve Streetprint, I realized that the time was also right to make good on my year-long promise of integrating some manner of metadata support into the Engine. After some initial research, we focused our efforts on mapping Streetprint-specific fields to their Dublin Core Metadata Initiative[22] (DCMI) equivalents, where applicable. DMCI's element set is an excellent standard and the current favourite in the digital resources metadata landscape, but we also architected our metadata implementation such that alternate mappings into other formats could be implemented with relative ease. For our initial implementation, we decided that site editors could enable or disable DCMI metadata on a site-wide basis; when enabled, DCMI-compliant META tags are included in the XHTML of each collection item's page. The Streetprint development team is already working a more robust method of exposing our object metadata via the XML schema recommended by the Open Archives Initiative[23].

---

[22] http://www.dublincore.org

Much to my relief, neither the metadata nor the user comment features proved very difficult to integrate into version 2.0's object-oriented architecture. More difficult to implement, however, was our deceptively-named "basic search" feature. The idea for "basic search" was a simple one: provide the ability for a user to search an entire Streetprint site's contents by entering their search query into a single text box, much like the familiar Google interface. Streetprint's existing search interface was powerful but perhaps overly complicated, consisting of a number of fields and checkboxes which forced users to specify exactly which combination of fields they wanted to search. This was ideal if you wanted to search *Revolution and Romanticism* for ballad sheets published between 1700 and 1730 which contained "sailor" in the title, for example. What we wanted to instead provide was a way to search simply for the word "sailor" and return results from *all* items which matched this search term in any way. This would become our new default search, and our existing search page would be rechristened "advanced search," since it permitted more fine-grained queries.

In a series of articles on the subject of online text searching, software developer Tim Bray notes that "weirdly enough, it's much easier to offer a good advanced than simple search" (1). We quickly discovered the truth of this statement; it is much easier to extract useful data from your database when you give it a very precise query—select only ballad sheets, only between a certain date range, and only with 'sailor' in the title—than when you supply the digital equivalent of "hello, Sailor!" To be more specific, we found that our database was very good at providing a list of collection items which matched the word "sailor." What it left to the software developer, however, was the delicate matter of *ordering* these search results. Why does ordering matter? As Tim Bray explains, "most people, after they've done a search, won't look at more than one page of result list" (1). If a search query returns fifty (or fifty thousand) items and only twelve items fit on a page of search results, you want to ensure that the most important and relevant items appear on that first page. But when developing a software engine which can power a myriad of digital collections, how do you make decisions about which items or fields of (unknown) data are the most "important?"

---

[23] http://www.openarchives.org/

Not surprisingly, the Streetprint development team suddenly found itself back in the realm of programming politics. Once we hit upon a way for the database to enumerate which fields in the 'Texts' table matched a particular query's search terms—does "sailor" appear in the item's title, notes field, fulltext record, or all three?—it was left to me to decide how to best use this information in an ordering of search results. I soon settled on what appeared to be the only reasonable solution: I developed a weighting algorithm which assigned "points" depending on which fields matched each of the query's search terms. Each matching collection item received a total number of points based on this algorithm, and the items were then displayed in descending point order. Algorithmic politics can hardly become more explicit; after deciding that a match in the "title" field was worth twelve points and an "author" match worth six, I recall sitting paralyzed in front of my computer monitor. Thoughts raced through my head: *what is more important, the fulltext record or the document type? City or publisher? I don't even know what users will put in these fields, why am I making these decisions? And will anyone who uses the basic search feature even notice I've made them?* I consulted the rest of the development team to see if we could build a tool into the Streetprint back-end which would allow site editors to specify their own weighting criteria, emphasizing the fields they felt were most important to their collection. We ultimately agreed that while such a feature would be an excellent future addition for "power user" site editors, it presented an interface design challenge, it would likely confuse novice users, and it simply couldn't be implemented anytime soon. Instead, I made decisions based on what would work best for our flagship site, ***Revolution and Romanticism,*** and then abstracted my basic search weighting algorithm into a separate function and placed it at the very top of the Engine's basicsearch.php file. If anyone asks, I thought to myself, I'll tell them exactly how to modify it. "Here's the code," I'll say. "It's your site. Make it work for you."

Version 2.1 of the Streetprint Engine was released on April 23, 2004, incorporating our new metadata, user comments, and basic search features as well as a number of smaller bug fixes, interface enhancements, and an automated upgrader for our existing 2.0 users. Arriving only six months after version 2.0, the new version was 2438 lines of code and 18 files larger than its predecessor. Moreover, I felt that the release of version 2.1 represented a vindication of our development model and our new object oriented architecture. In less than half a year, our three-person development team (all of whom worked only part-time on the Streetprint Engine) had managed to address a number of criticisms and incorporate a wide variety of improvements into the software, without sacrificing backwards-compatibility (leaving our existing sites stuck in version 2.0) or our core values.

Arkives and the Future of Streetprint

After reflecting on Streetprint's successes and failures to date, Gary Kelly decided in the spring of 2004 that the project needed a revised mandate and a new initiative. We had created a powerful tool for showcasing digital collections of texts and other artifacts, and we were starting to gain recognition in both the academic and online worlds. The problem, Kelly argued, was that we had largely been preaching to the converted; what we really needed to do was bring awareness of Streetprint to people and groups for whom the concept of creating a viable digital archive was inaccessible, even immaterial. The primary advantages of the Streetprint Engine were its ease-of-use and its low cost. The outlay associated with hiring consultants to build a custom website and content-management system could typically run into the tens of thousands of dollars. Since our software was free, all you needed to start a Streetprint site was a minimum of technical skill, a personal computer, a scanner or a digital camera, and a (cheaply available) web host. Material archives are by no means limited to academia. What if the world's non-profit organizations, grassroots political groups, religious groups, art galleries, and community leagues could use the Streetprint Engine to preserve and share their primary materials and printed ephemera with each other and the world? We quickly realized that Streetprint would require very little customization in order to become useful to these communities—in fact, by providing these groups with our "academic strength" software, Streetprint could serve as a great equalizer, reminding everyone that research is hardly limited to university settings.

Kelly has dubbed this idea the "Arkives Initiative," and this project will eventually grow to encompass a wide range of community archives powered by the Streetprint Engine, providing a non-academic (though certainly no less valuable) counterpoint to *Streetprint.org*. Some websites which will become part of the Arkives Initiative are already underway. Kalin Jensen's *Edmonton Activist Literature*[24] uses the Engine to showcase and preserve the ephemeral print of Edmonton's vibrant activist community. Lwam Ghebrehariat's *Eritrean Print and Oral Culture*[25] houses a wide variety of texts and artifacts which form a part of Eritrean culture, incorporating audio and video clips of traditional folktales, cultural ceremonies, and more. Our largest Streetprint site to date will be launched in Winter 2005, showcasing thousands of artistic works and pieces of printed ephemera from the Owens Art Gallery in Sackville, New Brunswick.

Back in the CRC Studio, an expanded development team is now hard at work on improving the Streetprint Engine and ensuring it meets the needs of our new audiences. Daria Kawecka, a recent University of Alberta computing science graduate, has been re-architecting key parts of the Engine's code, applying OOP principles to areas—such as user interface elements and our file structure—which we missed during the development of version 2.0. John Komick, a computing science student with a background in history and philosophy, is developing a "Streetprint Image Application," the CRC Studio's first foray into desktop application software. Intended for use in Winter 2005, the image application will provide a basic set of image editing tools as well as features which automatically enhance sets of images, create multiple resolutions, and then upload them directly to a Streetprint site. This free alternative to programs such as Adobe® Photoshop® will eventually be included with the Engine itself, helping to cement our software's low cost and ease-of-use.

---

[24] http://www.crcstudio.arts.ualberta.ca/activist/

[25] http://www.crcstudio.arts.ualberta.ca/eritrean/

Many of the next generation Engine features currently in development are built atop emerging technological standards. Our new "multi-site search" feature—which queries multiple Streetprint sites from a single location and displays results from across different collections—will be built using SOAP (Simple Object Access Protocol) web services, an XML-based protocol for the exchange of messages. Our upcoming metadata implementation follows the guidelines of the OAI (Open Archives Initiative), allowing Streetprint sites and their contents to be deployed as a black box within even larger online databases of cultural material. Our new website layouts will be coded entirely in CSS, an ideal way to separate the form of a webpage from its content, making it even easier for users to customize the look and feel of their Streetprint sites. Because the black-boxed power of technological standards like these provide us with ready-made solutions to problems, we can now devote greater amounts of time to ***researching*** our problems and the architectures we can deploy to solve them. In this way, we are increasingly able to avoid the do-it-yourself impulses of "Not Invented Here" syndrome. By the same token, when the available technological architectures fail to meet our unique needs as literary researchers and humanist digital archivists, we can refashion these technologies and standards in our own image, forging a path for new kinds of digital research. What's more, the politics behind our programming—the decisions being made every day by the Streetprint development team—are now publicly accessible; everyone is welcome to join in the growing conversation on our developer forums[26] or to download our source code.

Laurence Lessig reminds us: "nature doesn't determine cyberspace. Code does. Code is not constant. It changes...how it changes depends on the code writers. How code writers change it could depend on us" (*Code* 109). Like any software project, the Streetprint Engine will never be "complete." In the past two years, Streetprint has grown from a tiny, single-collection system into a powerful set of tools for powering digital collections of popular print, printed ephemera, and more. Along the way, we have attempted to change many of the rules governing the use of digital technology in the humanities. Software need not be expensive. Software need not be impersonal, insidious, or ineffable. Software should be transformed by its users, and not the other way around. Software can be beautiful. No software is an island; software can (and should) communicate in surprising ways. Software is powered by code, and code is not constant. This chapter has traced the story of the Streetprint Engine and its development by an unconventional team of code writers. In the years to come, how will these code writers change it? That could depend on *you*.

---

[26] http://www.crcstudio.arts.ualberta.ca/streetprintorg/forums/index.php?board=14

Chapter 2

Hackers, Humanists, Intellectuals: Lessons from the Open Source Revolution[26]

> I want to argue that in these societies in which the knowledge mode of production prevails...intellectuals may become the only genuine political class.
> — Stanley Aronowitz, "On Intellectuals", 52

> In our society, the empowered intellectuals tend to be not cultural intellectuals but their colleagues among scientists and engineers.
> —John Michael, ***Anxious Intellects***, 13

> Events led me to ask myself, as a software designer, what I should do with the software I develop in order to benefit humanity the most. In particular, I asked the question of whether it was ethical to make software proprietary...we must ask what we, as programmers together, can do for the freedom of mankind. We must ask what we ought to do, not just what is profitable.
> — Richard Stallman, "Why Software Should be Free", 190

The last two decades have witnessed a mounting sense of crisis among humanities scholars and the "academic Left" in general. While the opinions of such academics have always had detractors, it has become increasingly clear that the humanist mode of knowledge production itself—manifested in the kind of self-directed labour that Stanley Aronowitz termed "The Last Good Job in America"—is also under siege. Indeed, the humanities (and humanities workers) face a daunting array of opponents: global capitalism and the increasing corporatization of universities, the disappearance of traditional sources of funding, attacks from colleagues in the sciences ("science wars"), and internal squabbling have all contributed to the weakening and fragmentation of

---

[27] An earlier version of this chapter has been published in the proceedings of the "Culture & The State" conference at the University of Alberta. Volume 4. Eds. James Gifford and Gabrielle Zezulka-Mailloux. Edmonton: CRC Studio, 2003. 24-43.

humanities departments. Although many voices have spoken out against these developments and urged a unified resistance, some critics have already declared "the retreat of the humanities from the line of intellectual and political resistance" (Miyoshi 39) a *fait accompli*. Others—most notably Bill Readings in *The University in Ruins*—have argued that the "presumption of a shared or common humanity is an irresponsible desire" (189) given the implications of postmodern theory, and that a "community of dissensus" is instead the closest thing to "unity" towards which the humanist academy (or universities as whole) should strive.

As a student with dual allegiances to my university's English and Computing Science departments, I have always felt—even prior to my experience with the Streetprint project—that visions of interdisciplinarity need not be so pessimistic. Yet in light of cross-disciplinary battles such as the infamous "Sokal hoax" of 1996, many feel Readings' advice to be pragmatic as well as theoretically sound. While dissensus might be a productive model for coping with interdisciplinary heteronomy, I would like to suggest that valuable alliances can still be forged and important lessons learned by exploring interdisciplinary similarities. The last few years have indeed seen increasing attempts to begin healing the humanities-sciences rift. Unfortunately, many critics attempt to accomplish this not by breaking down the oppositional humanities vs. sciences binary but by simply re-aligning the humanities and the "hard" sciences against the applied sciences:

> The so-called hard sciences, for the first time in fifty years, find themselves lumped with the humanities as less important and less well funded departments in universities increasingly devoted, even at the undergraduate level, to programs in engineering and management. (Michael 152)

Michael accurately captures the economic and political hardship which is now increasingly shared by humanities departments and the hard sciences alike, and rightly condemns the "science wars" as unproductive and divisive. Those among the humanities and the academic Left searching for "spiritual allies" and sites of political resistance—both inside and outside the university—would be unwise to simply ignore everything beyond the borderline between "hard" and "applied" science, however. Unbeknownst to many humanities scholars, the past twenty years have seen the remarkable growth of an idealist technological and political movement which has changed the shape of global communication and become the largest single threat to one of the world's most powerful corporations, all while promoting values which are surprisingly resonant with those of the academic Left.

Founded by Richard Stallman in 1984, in the last two decades the Free Software / Open Source[28] movement has emerged from university Computing Science departments and technical schools to become a largely autonomous political group whose inspiring history may hold lessons for the humanist academy. During my experience over the past two years developing the Streetprint Engine, I was consistently surprised at how often the values and methodologies of this group corresponded to my own. As humanists become increasingly involved in the process of crafting digital archives and other technological architectures for communicating and preserving "the entirety of our received cultural archive of materials" (*Radiant Textuality* 169), an understanding of the history and practice of the open source movement is crucial. If nothing else, even the most technophobic of humanities academics should be pleasantly surprised to discover just how much they share in common with a group situated firmly within the applied sciences: computer hackers.

---

[28] I use the terms "Free Software" and "Open Source" together here because they are often used interchangeably. As I will explain later, however, the two are not entirely synonymous.

Before delving into the history of the Free Software / Open Source movement, it is worthwhile to examine the shifting signification of the term "computer hacker". Usually shortened simply to "hacker", the term is problematic due to its unfortunate historical associations with activities such as virus-writing and illegal break-ins to computer systems. The term originated in the 1960s as a badge of honour among scientists at MIT Artificial Intelligence lab, but by the 1980s the term had become widely associated with computer criminals, propelled mostly by sensationalistic media reporting which served to glamourize hacker activities as romantic countercultural tendencies (Ross 84). While this sense of "hacker" still persists in circulation today, hackers themselves have been working for over a decade to reappropriate the term and return its original, more positive connotations. Hacker activist Eric Raymond's online "Jargon File" does an excellent job of delineating the term's "official" modern usage:

> **hacker** n.
> [originally, someone who makes furniture with an axe] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. 3. A person capable of appreciating **hack value**. 4. A person who is good at programming quickly. 5. An expert at a particular program, or one who frequently does work using it or on it; as in `a Unix hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.) 6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example. 7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations. 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence `password hacker', `network hacker'. The correct term for this sense is **cracker**.
> (http://www.tuxedo.org/~esr/jargon/html/entry/hacker.html)

By offloading the more sinister connotations of "hacker" onto the more appropriate term "cracker", hackers can once again use the term for positive communal self-identification. The expansion of the term to include experts in fields other than software engineering (definition 6, above) is also an interesting move towards community-building. In his essay "How to Become a Hacker," Raymond further develops this idea, noting

> the hacker mind-set is not confined to this software-hacker culture. There are people who apply the hacker attitude to other things, like electronics or music—actually, you can find it at the highest levels of any science or art. (196)

Initial comparisons of hackers to humanities academics, based upon this definition, might appear superficial. ~~It~~ could be pointed out that both groups apply craftsmanship to create "texts" (whether in natural or programming languages), or one could resort to platitudes and claim that both hackers and academics are "doing it for love." While this may certainly be the case, more significant sonorities emerge once hacker impulses are applied to conceptions of labour. In "The Last Good Job in America", Stanley Aronowitz writes that

> backbreaking manual labor and work that separates body and mind should be eliminated by technology and democratic work organization. And rather than excoriate those who are able to avoid the most degrading jobs, the Left should excoriate those jobs and not romanticize them. In a word, most paid labor is shit, and those who are lucky enough to avoid it, make it more pleasant and self-managed, or reduce it to the barest minimum should be emulated. (211)

Aronowitz need look no further than hackers to find an example worth emulating. In his excellent book *The Hacker Ethic*, sociologist Pekka Himanen assesses the hacker work philosophy as follows:

> The hacker view is that the use of machines for the optimization and flexibility of time should lead to a life for human beings that is less machinelike—less optimized and routine. Raymond writes: "To behave like a hacker, you have to believe this [that people should never have to drudge at stupid, repetitive work] enough to want to automate away the boring bits as much as possible, not just for yourself but for everybody else." (33)

The similarities are both obvious and startling, especially since most humanities technophobes might traditionally suspect the opposite of hackers—that they aim to make human labor *more* machinelike. On the contrary, hackers understand just as well as academics that personal control over one's work time is crucial to the creative process. Himanen explains that

> a hacker may join his friends in the middle of the day for a long lunch or go out with them for a beer in the evening, then resume work late in the afternoon or the next day. Sometimes he or she may spontaneously decide to take the whole day off to do something completely different. (33)

Once again, there are striking parallels with Aronowitz's description of his "last good job in America:"

> What I enjoy most is the ability to procrastinate and control my own work time, especially its pace: taking a walk in the middle of the day, reading between the writing, listening to a CD or tape anytime I want, calling up a friend for a chat. (207)

In the "struggle for universalizing the self-managed time some of us still enjoy" (Aronowitz 221), academics and hackers are clearly fighting on the same side. The question must be asked, however: is the hacker work ethic simply an idealist philosophy which looks good on paper? How have hacker tenets played out in the "real world?" To begin answering this question, we must turn to "the closest thing to a leader of the hacker culture" (Raymond 69) and examine the history of uber-hacker Richard Stallman and the movement he founded in 1984.

Richard Matthew Stallman was born in New York in 1953. An only child, his father owned a printing company and his mother was a teacher (Moody 14). He began studies at Harvard in 1970, and graduated there with a Physics degree in 1974. Despite his official schoolwork, however, he was already working as a programmer at the famous MIT Artificial Intelligence lab by 1971, and he would stay there for the next ten years. The 1970s were a heyday for academic computing science. As Charlie Lowe explains,

> in the 1970s, computer programmers enjoyed a freedom to use their texts—the software that they write—that teachers and researchers in other disciplines would imagine as nearly ideal. Funded extensively by corporate sponsorship, software produced by programmers in the academy was freely shared in the US in exchanges across the academic community. (1)

This freedom of sharing meant that no two hackers ever had to duplicate the same work; the source code of others—the lines of text which tell a program how to do what it does—was always available to learn from, build upon, and improve, helping advance the field of computing science theory and providing obvious benefits to programmers and "end users" alike. It was not long, however, before things began to change. In the introduction to his early essay "Why Software Should Be Free," Stallman wrote:

> In the time that I've worked as a programmer, I've watched the field change from one of cooperation and sharing, where people could reuse previous work to advance the state of the art, to one in which cooperation is largely forbidden by trade secrecy and sharing is illegal. (190)

This change occurred because as personal computers became more and more popular towards the end of the 1970s, corporations discovered that software could be packaged and sold at handsome profits, especially since once development was complete, duplicating the end product (the computer program itself) was practically costless. (Prior to this "discovery," hardware sales were the primary source of income for the vast majority of computer companies.) Of their academic sponsorship,

> corporate sponsors realized that the software they were paying for was valuable intellectual property to be protected, not shared. The corporations began copyrighting anything their sponsorship produced, restricting the right of programmers to release their code and build on the code of others. (Lowe 1)

This situation is largely the status quo today. When you purchase a copy of Microsoft Word, for example, there is no way to figure out how the program does what it does; you cannot add a feature to it, fix a bug, base a new program upon it, or (if you're a non-programmer) hire your favourite programmer to do any of these things. It was against this new status quo that Stallman decided to protest in the only way he could: by hacking.

Gramsci writes that "every social group...creates together with itself, organically, one or more strata of intellectuals which give it homogeneity and an awareness of its own function not only in the economic but also in the social and political fields" (5). Stallman's aims have always been social and political; in his characteristically blunt fashion, he explains that "for the free software movement, non-free software is a social problem and free software is the solution" ("Why 'Free Software' is better than 'Open Source'" 1). It is hardly a stretch to imagine Stallman as a Gramscian "organic intellectual." More than any other single figure, Stallman has worked to encourage computer hackers to recognize their work ~~in a~~ broader social context. In 1984 he established the Free Software Foundation (FSF) as a charitable organization to help evangelize his belief that software and source code should have no owners. The word "free" in "Free Software Foundation" does not refer to price, but to freedom. In his classic example, Stallman notes that

> ``Free software'' is a matter of liberty, not price. To understand the concept, you should think of ``free'' as in ``free speech,'' not as in ``free beer.'' ("The Free Software Definition" 1)

Himanen points out that "to many, this may initially sound like a form of communism or even utopianism" (58). But Stallman's distinction between "free speech" and "free beer" makes his point a bit more subtle. Software, manuals, technical support and other services can still be sold in Stallman's model as long as the source code to the software is always provided without restriction. Stallman argues persuasively that the harm done to society by obstructing software vastly outweighs the economic benefit of hoarding the source code and selling it at a profit. Furthermore, he expounds his belief that "programming is fun" and that most hackers would continue to program for love, not money. Stallman compares programming before proprietary software to

> other fields of study and art in which there is little chance to become rich, which people enter for their fascination or their perceived value to society. Examples include mathematical logic, classical music, and archaeology; and political organizing among working people. People compete, more sadly than bitterly, for the few funded positions available, none of which is funded very well. They may even pay for the chance to work in the field, if they can afford to. ("Why Software Should Be Free" 196-7).

Stallman then points to the many institutions which funded software development throughout the 1970s—such as universities and hardware companies—and insists that they could still provide similar degrees of funding today.

While the birth of the free software movement in 1984 was a turning point for Stallman, it did not mark a shift from programming towards political activism; he instead pursued both activities with unprecedented intensity. The standard operating system for commercial and institutional computing at the time was known as "Unix," and the majority of its components had become proprietary trade secrets by 1984. In the face of this development, Stallman approached the problem in typical hacker fashion: he decided to code his own Unix "work-alike," which would duplicate every component and function of a usable Unix system but would do it with free software. He dubbed this project

"GNU," an acronym which stood for "GNU's Not Unix."[29] Since Unix was composed of many small pieces which together functioned as a complete operating system, Stallman got to work writing "small pieces" of code at his MIT office, where the university continued to let him work (and even sleep)! Moody notes that "although this makes the entire process sound disarmingly simple, writing these 'small pieces' involved reproducing work that had taken hundreds of people fifteen years" (20). Stallman spent the next several years mostly working alone, consumed by his task. He also began to prove that a programmer could make a living writing free software; by 1985 he was earning his keep through consulting work, making custom modifications to (and selling tapes of) free software he had created such as GNU Emacs, a text editor which is still a standard today.

One of Stallman's greatest innovations was not a piece of programming code, however, but a piece of text. With the help of a lawyer, in 1985 he drew up the first version of the "GNU General Public License," or GPL. Stallman began including this license with all the programs released by the Free Software Foundation, and he encouraged all other programmers interested in writing free software to do the same. As Lowe explains,

> when software is copyrighted and published under the GNU GPL, not only is the source code provided, but the owner of a piece of software is given the rights to modify, copy, and redistribute it. The license does carry one significant restriction: ***any new version or copy must be published itself under the GNU GPL***, thus guaranteeing that any improvements on the original code would continue to be freely available. (1, emphasis mine)

This "one significant restriction" of the GPL was Stallman's master stroke, an "extraordinarily clever approach to propagating freedom" (Moody 27). It meant, for example, that if a proprietary software company wanted to incorporate GPL code into a software product—which, by the conditions of the GPL, would be their right—they would then have to release the source code to their *entire* program under the GPL. This "viral" approach to freedom enshrined in the GPL in some ways constituted a more valuable and important "hack" than any of Stallman's software projects.

---

[29] The acronym betrays typical hacker humour: the first word of the acronym is of course self-referential, mimicking a popular programming technique called "recursion". PHP, the programming language used in the Streetprint Engine, also boasts a recursive acronym: "PHP: Hypertext Preprocessor."

By the end of the 1980s, the Free Software Foundation had grown from a one-man operation to a shop of nearly fifty programmers (Moody 20) funded mostly through private donations and the occasional grant from governments and universities. The GNU project's goal—the creation of a completely free operating system which duplicated every function of the best commercial Unix systems—was nearly complete. As well, many programmers not affiliated with the FSF or the GNU project had begun to publish their code under the GPL and to make improvements to GNU programs. But one thing was missing from GNU's complete working system: a free kernel. As the name suggests, a "kernel" sits at the center of a computer operating system. Moody explains that the kernel "acts as housekeeper (handling tasks such as the screen and keyboard) and judge (ensuring that all the competing calls for attention from user programs are handled correctly)" (25). One of the reasons that Stallman had not yet developed a free kernel was that a working kernel is a prerequisite for almost all programming activities, including kernel programming itself. Since all GNU programs conformed to Unix specifications, they could run on top of any of the proprietary Unix kernels which were still in wide use at the time. Stallman had made plans to develop a GNU Kernel (called the "Hurd") to realize the last step of his dream for an all-free system, but the task was complex and a working version was still many years away.

Across the Atlantic in 1991, a young Finnish computing science student named Linus Torvalds had an idea. He had been introduced to Unix in his classes that year, and liked it immensely. The only computer that ran Unix at Helsinki University supported just 16 users at a time, however, and so Linus was frequently frustrated by having to wait in line to complete assignments. Moody tells the story of a discussion which took place that year between Torvalds and a fellow classmate:

> "One of the things that we discussed [was that] we might need to actually write [a] Unix ourselves because the commercial ones were really expensive," Wirzenius says, but adds, "that was a joke, especially from my side; I'm not quite as good a programmer as Linus is. It's not something you decide–'this week I will write an operating system.'" The jest proved truer than either could have imagined. (32)

Once he realized that writing his own kernel would allow him to run a Unix-like system for free on his home PC—and possibly use the wide variety of GNU applications which had become freely available by this time—Torvalds decided that his project was actually worth attempting. In September 1991, Torvalds did something incredibly prescient: he posted an extremely early, primitive version (version 0.01) of his new kernel—which by now had been dubbed "Linux" by his friends—on an internet site and announced the availability of the source code in several hacker newsgroups. The internet had just begun to hit its stride in the early 1990s, although it was still primarily used by hackers and academics. The reaction to Torvalds' internet posting was immediate and overwhelming: he quickly began receiving bug fixes, additions, and feedback as the hacker community became interesting in turning Linux into a complete operating system. True to his motto of "release early, release often," Torvalds began posting new versions of Linux (which he decided to publish under the GPL) almost nightly, and the Linux developer community grew quickly. Another Torvalds truism was that "given enough eyeballs, all bugs are shallow," which helped explain how Linux was able to improve so quickly; any bugs or flaws in the code were usually spotted quickly by another hacker and fixed with incredible speed. Eric Raymond helps explain part of the appeal of the Linux project to hackers:

> Linus was keeping his hackers/users constantly stimulated and rewarded—stimulated by the prospect of having an ego-satisfying piece of the action, rewarded by the sight of constant (even *daily*) improvement in their work. (30)

Within the academy, however, many hackers and computing scientists were skeptical that the Linux development model could truly lead to well-designed, powerful software. In his seminal essay "The Cathedral and the Bazaar," Raymond describes the gulf between the Linux model and more traditional models of software development. Conventional wisdom dictated that really important software—such as operating systems or large projects like Stallman's Emacs text editor—needed to be built "like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta to be released before its time" (21). By way of contrast,

> Linus Torvald's style of development—release early and often, delegate everything you can, be open to the point of promiscuity—came as a surprise. No quiet, reverent cathedral-building here—rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, which would take submissions from *anyone*) out of which a coherent and stable system could seemingly emerge only by a succession of miracles. (21-22)

Amazingly, it worked, and worked well. By leveraging the strengths of the emerging internet and maintaining a relatively hands-off approach, Torvalds was able to maintain Linux's high quality and pace of development even as the ranks of Linux developers swelled into the thousands. By the mid-nineties, the "open-source software movement" (as Linux users has taken to calling themselves) had grown into a worldwide phenomenon and spawned hundreds of other, non-Linux projects. Companies such as Apple, IBM, and Netscape began exploring the possible benefits of open-source code, and universities and governments also realized it was in their best interests to partially fund the development of Linux and other projects. One example of a successful open-source project is the Apache web server, which as of July 2004 powers an incredible 67% of all websites on the internet (Netcraft 1). In fact, the majority of the world-wide web was built (and is still powered) by open-source code. By late 1998, software behemoth Microsoft had even identified Linux as one of their most dangerous competitors. A leaked corporate memorandum known as the "Halloween Document" (later confirmed by Microsoft as legitimate) warned that

> OSS [Open-source Software] poses a direct, short-term revenue and platform threat to Microsoft, particularly in server space. Additionally, the intrinsic parallelism and free idea exchange in OSS has benefits that are not replicable with our current licensing model and therefore present a long term developer mindshare threat. (OpenSource.org 1)

Microsoft's post-1998 marketing attacks on Linux and open-source software have done relatively little to derail the movement. The Linux community is still growing today, and the Linux/GNU combination now has millions of users across the globe who conduct all their daily computing activities without a single piece of proprietary software.

In the opening paragraph of "The Cathedral and the Bazaar", Raymond writes:

> Linux is subversive. Who would have thought even five years ago (1991) that a world-class operating system could coalesce as if by magic out of part-time hacking by several thousand developers scattered all over the planet, connected only by the tenuous strands of the internet? (21)

While this does not overstate the commendable accomplishments of Linux, I am mindful of Terranova's warning against subscribing too completely to the "idealistic cyberdrool of the digerati" (44). It is not entirely fair to claim that the humanities academy is wrought with division and infighting whereas hackers have put aside all their differences and succeeded in challenging the status quo through their merry, communal unity and hacker work ethic. Raymond acknowledges the rather more complex realities of hacker culture in his essay "Homesteading the Noosphere":

> The ideology of the internet open-source culture (what hackers say they believe) is a fairly complex topic in itself. All members agree that open source...is a good thing and worthy of significant and collective effort. This agreement effectively defines membership in the culture. However, the reasons individuals and various subcultures give for this belief vary considerably (67).

Not surprisingly, the culture's most controversial figure is Richard Stallman. Himanen notes the prevalent opinion that Stallman's thinking is "so radical that many of the actual open-source companies prefer to keep their distance from him as a person" (58). In fact, the moniker "open source" (as opposed to "free software") became popular in the mid-nineties with those who were reluctant to be associated with Stallman's overtly social and political ethic, especially those who wanted to pitch open-source software to business. Stallman also has the ability to "offend even supporters through his constant refusal to compromise" (Moody 29). Moody quotes Stallman on this subject:

> "The only reason we have a wholly free operating system is because of the movement that said we want an operating system that's wholly free, not 90 percent free," he says. "If you don't have freedom as a principle, you can never see a reason not to make an exception. There are constantly going to be times when for one reason or another there's some practical convenience in making an exception." (29)

Stallman has rather famously turned down invitations to speak at Linux conferences due to the fact that he believes everyone should refer to the system as "Linux/GNU" instead of "Linux," thereby acknowledging the fact that Linux is technically just a kernel and software which the GNU project had been working on since 1984 constitutes a major part of the system. On his personal website, Stallman recommends against using the word 'open' as a substitute for 'free software', explaining that "a different group, whose values are less idealistic than ours, uses 'open source' as its slogan" ("Some Confusing or Loaded Words..." 1). Stallman correctly assesses the key difference between the two movements:

> For the Open Source movement, the issue of whether software should be open source is a practical question, not an ethical one. As one person put it, 'Open Source is a development methodology; free software is a social movement.' ("Why 'Free Software' is better than 'Open Source'" 1)

For many open source advocates, Stallman's "all or nothing" approach goes too far. These pragmatic enthusiasts prefer open over closed-source code when the bazaar development model suits the task at hand, but are willing to use and support both kinds of software. As with any large community, there is also a variety of positions located on the spectrum between these two poles of idealism and pragmatism. Some of these positions are represented by additional open source software licenses—such as the BSD license—which differ slightly from Stallman's GPL.

Also problematic are attempts to assess the "real world" success of the hacker work ethic. Terranova correctly notes that the open source movement relies strongly upon "the free labor of internet tinkerers" (49). While many leaders of open source projects are either academics or are supported by open source consulting companies and organizations such as the Free Software

Foundation, the majority of open source programming work remains unpaid. Since money is not the primary motivator, the hacker drive in practice is often attributed to hacker "gift culture." Raymond compares this gift culture to academic tenure, explaining that

> within it, there is no serious shortage of the 'survival necessities'—disk space, network bandwidth, computing power. Software is freely shared. This abundance creates a situation in which the only available measure of competitive success is reputation among one's peers. (81)

Of course, most hackers enjoy the luxury of guaranteed "survival necessities" not due to institutional sponsorship (as with academic tenure) but simply because programmers are economically valuable commodities. Terranova points this out, noting that "the internet is always and simultaneously a gift economy *and* an advanced capitalist economy" (51). While a computer hacker and a graduate student in history may both be "doing it for love," the latter is likely much more worried about finding a steady paycheque than the former. Ironically, proprietary software companies may in fact be subsidizing open source development, since their programmers can go home after work and "unwind" by coding free software projects. While the spirit of the hacker work ethic may indeed resonate with the ethic of the humanities academy and the academic Left, it is not surprising that hackers have had more success in living out their ideals; the current economy clearly favours knowledge workers who write in programming code over those who write in natural languages.

Despite this economic imbalance, the evolution of the open source movement over the past twenty years has nonetheless laid an incredibly fertile groundwork for digital projects in the humanities. Jerome McGann captures the incredible potential of such projects:

> We stand at the beginning of a great scholarly revolution. Even now we operate under the extraordinary promise this revolution holds out: to integrate the

> resources of all libraries, museums, and archives and to make those resources available to all persons no matter where they reside physically. ("Radiant Textuality" 381)

As we developed the Streetprint Engine in the CRC Studio, we faced several challenges: how could we build digital architectures that made our collection of popular print available "to all persons" while simultaneously honouring the nature of the original texts? How could we avoid planned obsolescence, and ensure that what was circulating freely within the digital commons would retain its freedom in the future? And how could we do all this within the budgetary constraints of a 21st century English department? The open source movement (and the products of its many revolutions, such as the GPL and the PHP programming language) supplied us with the technological, social, and legal means to meet each of these goals. Specific operating systems, data formats, and even programming languages come and go; by releasing the Streetprint Engine as open source code, we ensured the sustainability of our architecture—a crucial element for any archive, whether digital or analog, publicly or privately-funded. Future hackers will always be able to examine our source code and update it to fit their emerging needs, and the terms of the GPL ensure that this code will always remain available.

It is no coincidence that the open source movement can act as a powerful catalyst between tech and text. As technologies, the two are remarkably similar. Laurence Lessig makes this point in his book *Code*, noting that "books are open source software: they hide nothing; they reveal their source—they are their source!" (107). A book weaves its "architecture" through its bibliographical encoding and the words on the page; as with open source code, its supporting scaffolding is always immediately available to a reader/user. The Streetprint Engine is not alone in acknowledging this similarity. Many other recent experiments (and full-blown projects) by academics suggest that the humanities can take more than simple inspiration from their spiritual allies in the free software movement. Many scholars, for example, have drawn parallels between source code and the content of journals, noting that the free sharing of academic papers in research publications would hold benefits similar to those of free software. While internet publishing would appear to be an obvious forum for this sort of activity, Singer notes that

> because of their embargo policies, some prestigious journals (along with some not-so-prestigious ones) will not accept articles that have been published as electronic pre-prints on the internet. (2)

Several groups are working to counteract this tendency. Inspired in part by Stallman's GPL, Creative Commons[30] was founded in 2001 with the mission of injecting "balance, compromise, and moderation" into the debate over copyright and creative control. Since its debut, Creative Commons has developed

> a Web application that helps people dedicate their creative works to the public domain — or retain their copyright while licensing them as free for certain uses, on certain conditions. Unlike the GNU GPL, Creative Commons licenses are not designed for software, but rather for other kinds of creative works: websites, scholarship, music, film, photography, literature, courseware, etc. We hope to build upon and complement the work of others who have created public licenses for a variety of creative works. ("About Us" 1)

While Creative Commons can provide humanities researchers with open source-inspired licensing models, *Wikipedia.org* presents an astonishing example of open source development methodologies in action. Dubbed "the free encyclopedia," the site uses a system called Wiki to control its content. At first glance, Wiki seems to invite anarchy: any visitor to the site can, at the click of a button, edit the complete text of any of the entries in the encyclopedia, and the results are updated in "real-time." In abandoning any architected claims to authority—favouring instead the preservation of every version of every article and semi-anonymous, massive internet peer review—Wikipedia attempts to apply promiscuous, Linux-style bazaar principles as "purely" as possible to an encyclopedia project. Wikipedia has proved an incredible success. Instead of anarchy, there are an astonishing 300,000 articles in the English version of the website (as of July 2004), many of which are quite lengthy and of excellent quality. Remarkably, the open editing process seems to have encouraged collaboration, an impartial writing style, and the gradual improvement of article quality. Since Wiki also allows users to easily revert to previous versions of articles, any online vandalism which occurs is usually spotted and fixed almost immediately by Wikipedia users. While the project can, of course, never be "finished," Wikipedia is a fascinating application of open source principles, an engaging and entertaining online community, and an incredibly unique, useful humanities resource and archive.

---

[30] http://www.creativecommons.org

Led by Richard Stallman and other hacker intellectuals, Linux/GNU and its associated open source software projects have helped alter the shape of global communication and have emerged as a vital challenge to the status quo in the space of less than a decade. This astonishing progress is a testament to both the power of the hacker ethic and the collaborative potential of the internet. It is my hope that the Streetprint Engine will continue to draw upon—and contribute to—the incredible strength of this movement, placing the tools for using and creating open digital collections into the hands of many fellow researchers and the broader public. I also hope that Streetprint will not be alone for long. Differences remain between hackers and humanists, but as more and more researchers also pursue technological studies, increasingly larger segments of the humanities academy (and the broader community) will be able to take cues from the open source revolution and reap similar benefits. In particular, the architects of what McGann terms the "great scholarly revolution" of future digital archives must consider examining and adopting free software's values and practices; they owe it to their audience(s), to their material concerns, and to future generations. More generally, the history of the Free Software / Open Source movement can serve humanists as a striking example of the power of open community and collaboration between the arts and sciences, as one reminder that "totality is not always a monolithic system for the suppression of all differences and marginalities" (Miyoshi 42) but can instead, as I have learnt over the past two years, lead to very fruitful outcomes. Finally, humanities researchers should be heartened to learn that they may indeed have friends in the same place where, as an undergraduate arts student, I discovered a second home: the computing science department.

Works Cited

Aronowitz, Stanley. "On Intellectuals." *Intellectuals: Aesthetics, Politics, Academics.* Ed. Bruce Robbins. Minneapolis: University of Minnesota Press, 1990. 3-56.

——. "The Last Good Job in America." *Chalk Lines: The Politics of Work in the Managed University*. Ed. Randy Martin. Durham: Duke University Press, 1998. 202-221.

Bakken, Stig Sæther et al. "Appendix A: History of PHP and related projects." PHP Manual. 9 July 2004. <http://www.php.net/manual/en/history.php>

Barlow, John Perry. "Declaration of the Independence of Cyberspace." *Wired* 4.06, June 1996. 9 July 2004. <http://hotwired.wired.com/wired_online/4.06/declaration/>

Berard, Edward V. "Abstraction, Encapsulation, and Information Hiding." The Object Agency. 20 April 2003. <http://www.toa.com/pub/abstraction.txt>

Bray, Tim. "On Search: The Users." *ongoing*. 17 June 2003. 9 July 2004. <http://www.tbray.org/ongoing/When/200x/2003/06/17/SearchUsers>

Creative Commons. "About Us." *Creative Commons.* 9 July 2004. <http://creativecommons.org/learn/aboutus/>

Dahl, Ole-Johan and Nygaard, Kristen. "How Object-Oriented Programming Started." University of Oslo. 20 April 2003. <http://heim.ifi.uio.no/-kristen/FORSKNINGSDOK_MAPPE/F_OO_start.html>

Eliëns, Anton. "Paradigms of Programming." July 15, 2001. Vrije University, Course Materials. 20 April 2003. <http://www.cs.vu.nl/-eliens/oop/1-2.html>

Google.com. "Google Programming Contest." 2002. 20 April 2003. <http://www.google.com/programming-contest/winner.html>

Gramsci, Antonio. *Selections from the Prison Notebooks.* Trans. Quentin Hoare and Geoffrey Nowell Smith. New York: International Publishers, 1971.

Hasenbank, Andrea. "About Text in Transit." *Text in Transit.* 9 July 2004. <http://www.crcstudio.arts.ualberta.ca/textintransit/about.php>

Himanen, Pekka. *The Hacker Ethic and the Spirit of the Information Age.* New York: Random House, 2001.

Kelly, Gary et al. "About Cityprint." *Cityprint: Edmonton.* 9 July 2004. <http://www.crcstudio.arts.ualberta.ca/cityprint/about.php>

Latour, Bruno. *Science in Action: How to Follow Scientists and Engineers Through Society.* Cambridge: Harvard University Press, 1987.

Lessig, Lawrence. *Code and Other Laws of Cyberspace.* New York: Basic Books, 1999.

——. "The Code Is the Law." The Industry Standard. 9 April 1999. 9 July 2004. <http://www.lessig.org/content/standard/0,1902,4165,00.html>

Lowe, Charlie. "A Brief History of Open Source: Working to Make Knowledge Free." Kairos 6.2. Fall 2001. <http://english.ttu.edu/kairos/6.2/news/opensource.htm>.

McGann, Jerome. "Radiant Textuality." *Victorian Studies* (Spring 1996): 379-390.

——. *radiant textuality: literature after the world wide web.* New York: Palgrave, 2001.

——. *The Textual Condition.* Princeton: Princeton University Press, 1991.

Michael, John. *Anxious Intellects: Academic Professionals, Public Intellectuals, and Enlightenment Values.* Durham: Duke University Press, 2000.

Miyoshi, Masao. "Ivory Tower in Escrow." Boundary 2 27.1. 2000. 7-50.

Moody, Glynn. *Rebel Code: The Inside Story of Linux the Open Source Revolution.* Cambridge: Perseus, 2001.

Negroponte, Nicholas. *Being Digital.* New York: Knopf, 1995.

Netcraft. "July 2004 Web Server Survey." July 2004. <http://news.netcraft.com/archives/2004/07/01/july_2004_web_server_survey.html>

OpenSource.org. "Halloween Document 1." Version 1.14. 1998. <http://www.opensource.org/halloween/halloween1.php>.

Phoa, Wesley. "Should computer scientists read Derrida?" Sydney: University of NSW, 1994. 9 July 2004. <http://www.margaretmorgan.com/wesley/dcs.pdf>

Raymond, Eric. *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* Sebastopol: O'Reilly, 2001.

Readings, Bill. *The University in Ruins.* Cambridge: Harvard University Press, 1996.

Rose, Ellen. *User Error: Resisting Computer Culture.* Toronto: Between the Lines, 2001.

Ross, Andrew. *Strange Weather: Culture, Science and Technology in the Age of Limits.* New York: Verso, 1991.

SearchVB.com. "Pasta Theory of Programming." Whatis.com Definitions. 27 November 2002. 9 July 2004. <http://searchvb.techtarget.com/sDefinition/0,,sid8_gci866378,00.html>

Shepard, Leslie. *The History of Street Literature.* Newton Abbot: David & Charles, 1973.

Singer, Peter. "When Shall We Be Free?". Journal of Electronic Publishing. Volume 6, Issue 2. 2000. <http://www.press.umich.edu/jep/06-02/singer.html>.

Snopes.com. "Shoot Me Kangaroo Down, Sport." December 6, 1999. 20 April 2003. <http://www.snopes.com/humor/nonsense/kangaroo.htm>

Stallman, Richard. "Why Software Should Be Free." *Computers, Ethics & Social Values.* Eds. Deborah Johnson and Helen Nissenbaum. Upper Saddle River: Prentice Hall, 1995. 190-199.

—— et al. "Some Confusing or Loaded Words and Phrases that are Worth Avoiding." 2002. <http://www.fsf.org/philosophy/words-to-avoid.html>.

—— et al. "Why 'Free Software' is better than 'Open Source'". 2002. <http://www.fsf.org/philosophy/free-software-for-freedom.html>.

—— et al. "The Free Software Definition." 2002. <http://www.fsf.org/philosophy/free-sw.html>.

Suraski, Zeev. "The Object-Oriented Evolution of PHP." DevX.com. 9 July 2004. <http://www.devx.com/webdev/Article/10007/1954>

Terranova, Tiziana. "Free Labor: Producing Culture for the Digital Economy." Social Text 63. Volume 18, Issue 2. 2000. 33-57.

Ullman, Ellen. *Close to the Machine: Technophilia and Its Discontents.* San Francisco: City Lights, 1997.

Wikipedia.org. "Not Invented Here." 9 July 2004. <http://en.wikipedia.org/wiki/Not_Invented_Here>

Appendix A

This Appendix contains a series of planning illustrations, technical specifications, and screen captures related to the development of the Streetprint Engine, from it's initial version (1.0) through to our upcoming 3.x development version.

Pages 85–87 contain illustrations and database specifications for the initial 1.0 version of Streetprint, when the project was intended as simple, single-use website and database for British popular print.

Pages 88–104 provide the expanded database specification which accompanied the rechristened Streetprint Engine, version 1.5. Also provided are a number of screen captures from our first three websites.

Pages 105–109 illustrate two Streetprint websites, *Urban Record* and *Text in Transit.*

Pages 110–128 contain a series of object oriented planning illustrations loosely based on "Unified Modeling Language," or UML. These were used by the Streetprint 2.0 development team to aid the conceptualization of our new, object-based architecture. Also included is the complete 2.0 database specification, and screen captures of websites powered by version 2.0 of the Streetprint Engine.

Pages 129–131 illustrates the database specification for version 2.1 of the Streetprint Engine.

Pages 132–136 displays screen captures from the current version of *Streetprint.org*.

Finally, Page 137 includes a tentative SQL specifications related to the upcoming 3.x version of the Streetprint Engine, currently in development at the CRC Studio.

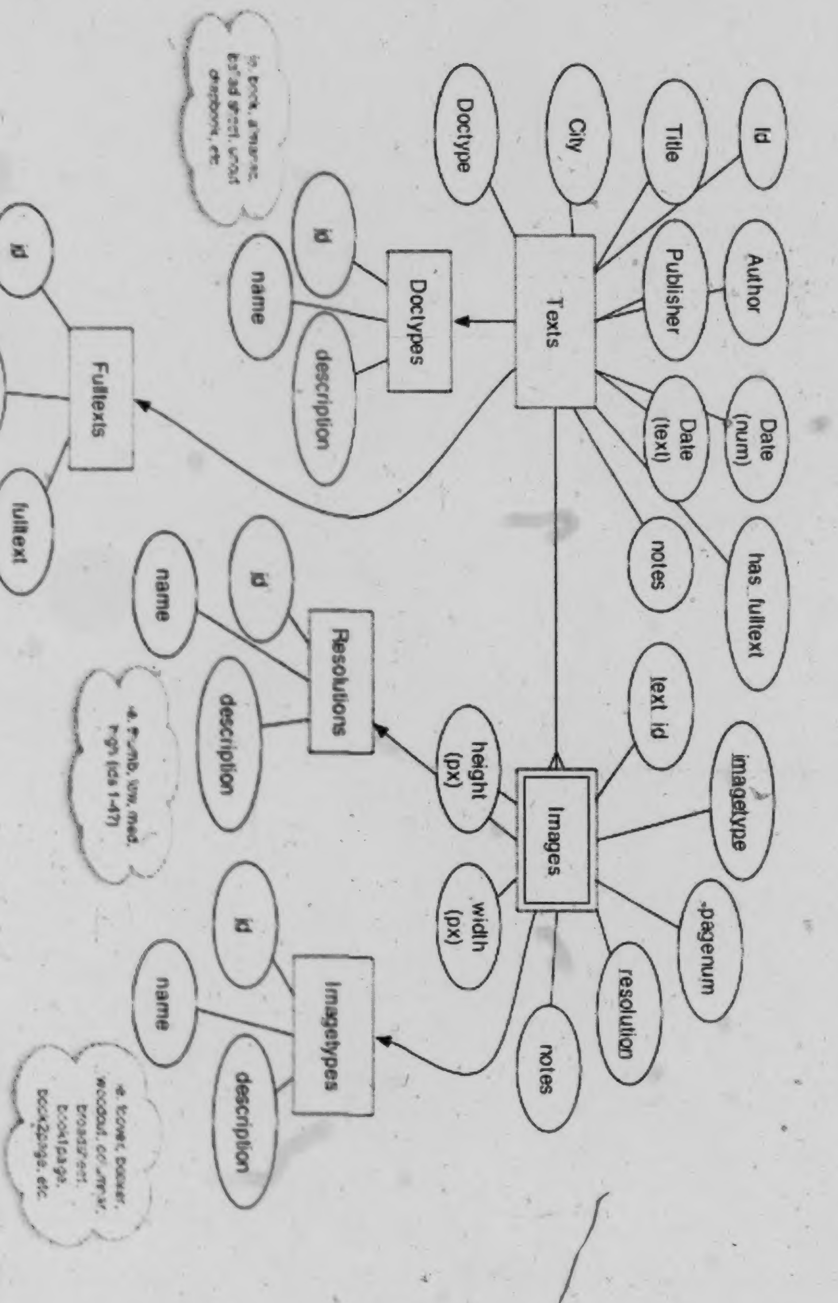
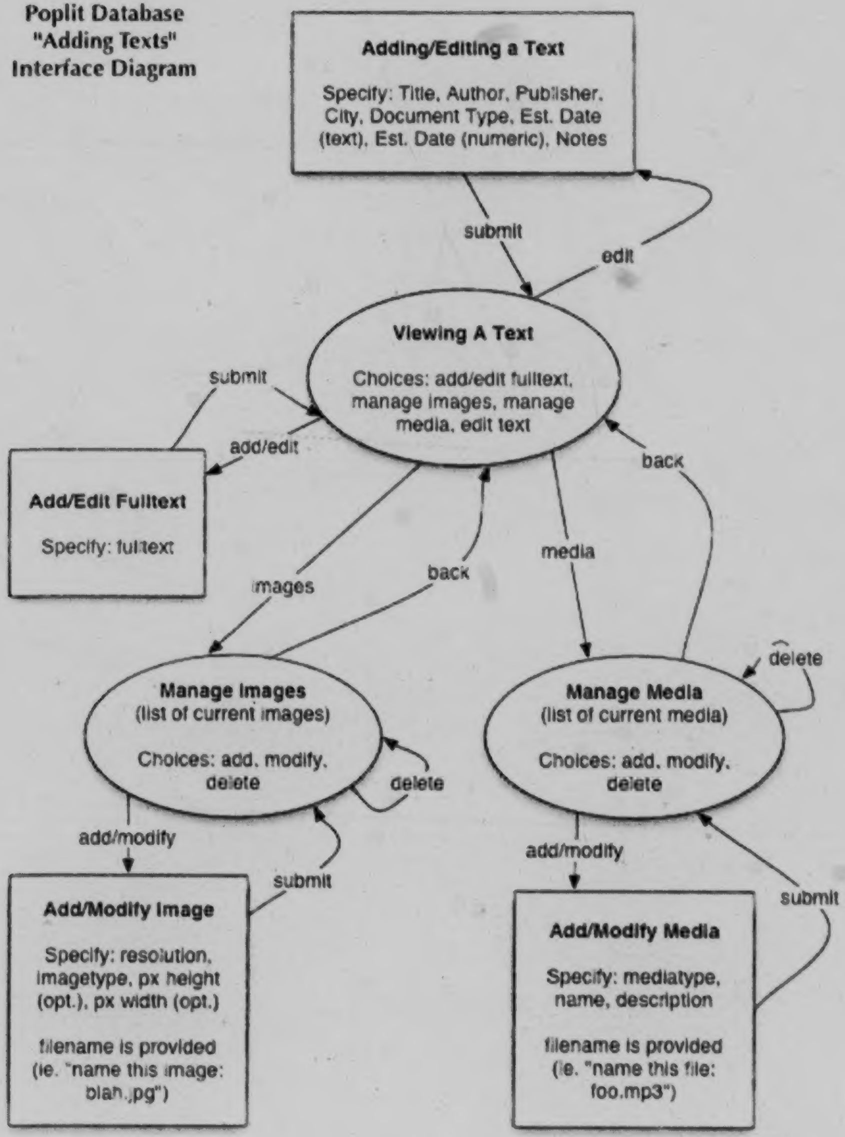

Texts
Id
Title
City
Doctype
Author
Publisher
Date (num)
Date (text)
has_fulltext
notes
Doctypes
id
name
description
ie. book, almanac, broadsheet, uncut drawings, etc.
Fulltexts
id
text_id
fulltext
Images
text_id
imagetype
pagenum
resolution
notes
height (px)
width (px)
Resolutions
id
name
description
ie. thumb, low, med, high (ids 1-4?)
Imagetypes
id
name
description
ie. cover, poster, woodcut, colophon, broadsheet, bookpage, book2page, etc.

Figure A-1
Version 1.0 Database Planning Illustration

```
CREATE TABLE texts (
id INT PRIMARY KEY auto_increment NOT NULL,
title TEXT,
author TINYTEXT,
publisher TINYTEXT,
city TINYTEXT,
doctype INT,
date_num INT,
date_text TINYTEXT,
notes TEXT);

CREATE TABLE doctypes (
id INT PRIMARY KEY auto_increment NOT NULL,
name TINYTEXT,
description TEXT);

CREATE TABLE resolutions (
id INT PRIMARY KEY auto_increment NOT NULL,
name TINYTEXT,
description TEXT);

CREATE TABLE imagetypes (
id INT PRIMARY KEY auto_increment NOT NULL,
name TINYTEXT,
description TEXT);

CREATE TABLE images (
text_id INT NOT NULL,
imagetype INT NOT NULL,
resolution INT NOT NULL,
height INT,
width INT);

CREATE TABLE fulltexts (
id INT PRIMARY KEY auto_increment NOT NULL,
text_id INT NOT NULL,
full MEDIUMTEXT);

CREATE TABLE media (
id INT PRIMARY KEY auto_increment NOT NULL,
text_id INT NOT NULL,
mediatype INT NOT NULL,
name TINYTEXT,
description TEXT);

CREATE TABLE mediatypes (
id INT PRIMARY KEY auto_increment NOT NULL,
name TINYTEXT,
extension TINYTEXT,
description TEXT);
```

Figure A-2

Version 1.0 Database Specification
MySQL "Create Table" Statements



Poplit Database "Adding Texts" Interface Diagram

Adding/Editing a Text
Specify: Title, Author, Publisher, City, Document Type, Est. Date (text), Est. Date (numeric), Notes
submit
edit
Viewing A Text
Choices: add/edit fulltext, manage images, manage media, edit text
submit
add/edit
Add/Edit Fulltext
Specify: fulltext
back
images
back
media
delete
Manage Images
(list of current images)
Choices: add, modify, delete
delete
Manage Media
(list of current media)
Choices: add, modify, delete
add/modify
submit
add/modify
submit
Add/Modify Image
Specify: resolution, imagetype, px height (opt.), px width (opt.)
filename is provided (ie. "name this image: blah.jpg")
Add/Modify Media
Specify: mediatype, name, description
filename is provided (ie. "name this file: foo.mp3")

Figure A-3

Version 1.0 Text Management Interface Diagram

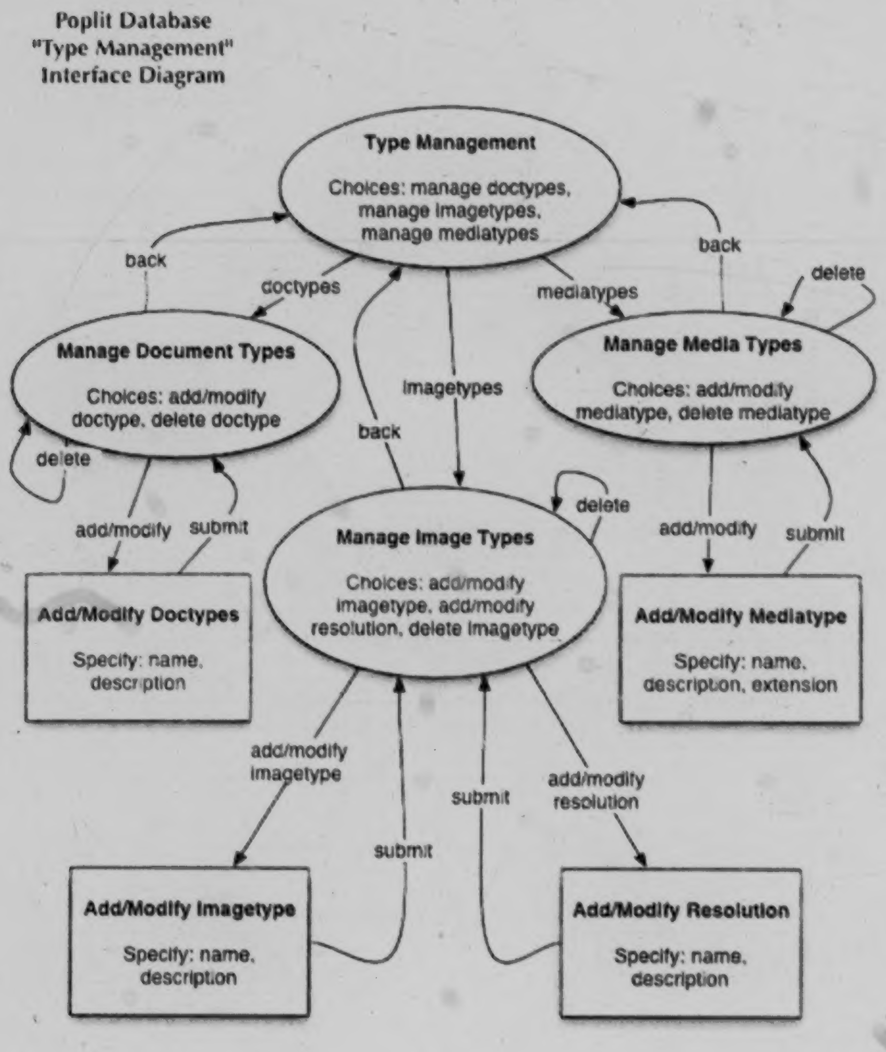

Poplit Database
"Type Management"
Interface Diagram
Type Management
Choices: manage doctypes, manage imagetypes, manage mediatypes
back
doctypes
mediatypes
back
delete
Manage Document Types
Choices: add/modify doctype, delete doctype
imagetypes
back
Manage Media Types
Choices: add/modify mediatype, delete mediatype
delete
delete
add/modify
submit
Manage Image Types
Choices: add/modify imagetype, add/modify resolution, delete imagetype
add/modify
submit
Add/Modify Doctypes
Specify: name, description
Add/Modify Mediatype
Specify: name, description, extension
add/modify imagetype
submit
add/modify resolution
submit
Add/Modify Imagetype
Specify: name, description
Add/Modify Resolution
Specify: name, description

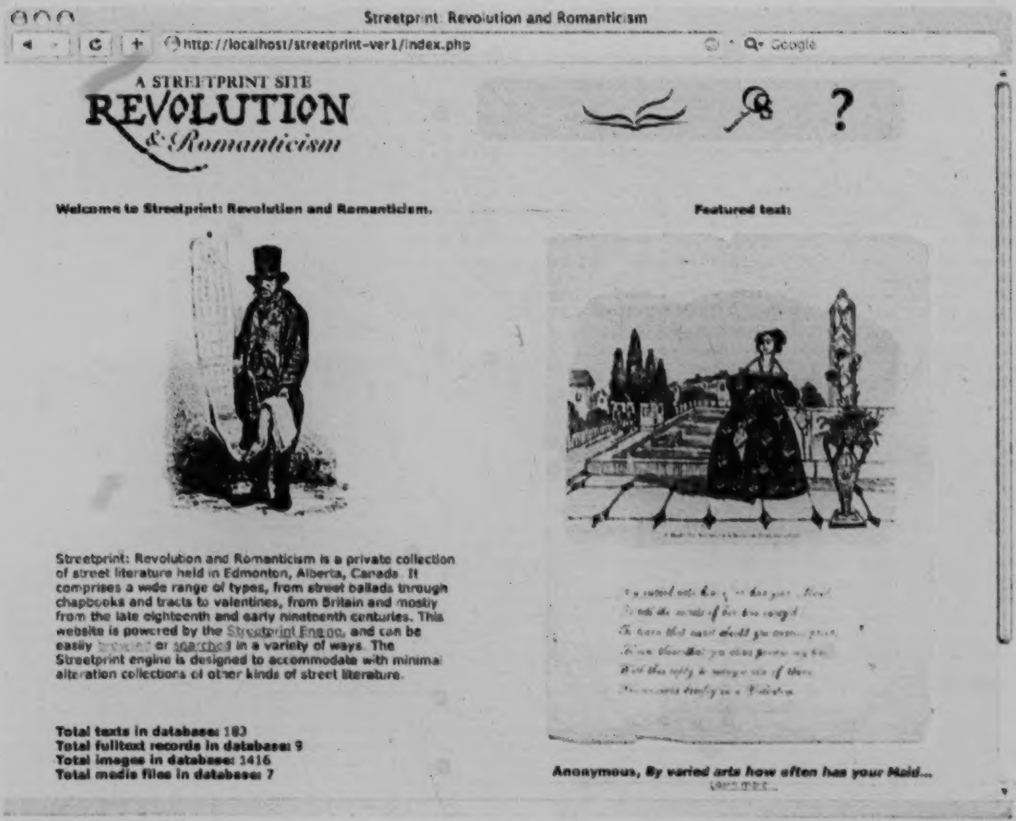
Figure A-4

Version 1.0 Type Management Interface Diagram

```
CREATE TABLE categories (
  id int(11) NOT NULL auto_increment,
  name text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE doctypes (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  def_imagetype int(11) default NULL,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE featuredtext (
  text_id int(11) default NULL
) TYPE=MyISAM;

CREATE TABLE firstlines (
  id int(11) NOT NULL auto_increment,
  text_id int(11) default NULL,
  firstline text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE fulltexts (
  id int(11) NOT NULL auto_increment,
  text_id int(11) NOT NULL default '0',
  full mediumtext,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE imageformats (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  extension tinytext,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

INSERT INTO imageformats VALUES (1,'JPEG','JPEG Compression, best for
photographs.','jpg');
INSERT INTO imageformats VALUES (2,'GIF','GIF compression, best for line art/vector
art.','gif');

CREATE TABLE images (
  id int(11) NOT NULL auto_increment,
  text_id int(11) default NULL,
  imagetype int(11) default NULL,
  resolution int(11) default NULL,
  caption text,
  format int(11) default NULL,
  firstpagenum int(11) default NULL,
  numpages int(11) default NULL,
  PRIMARY KEY  (id)
) TYPE=MyISAM;
```

```
CREATE TABLE imagetypes (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE media (
  id int(11) NOT NULL auto_increment,
  text_id int(11) NOT NULL default '0',
  mediatype int(11) NOT NULL default '0',
  name tinytext,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE resolutions (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

INSERT INTO resolutions VALUES (1,'Thumbnail','A small, minaturized representation of
an image.');
INSERT INTO resolutions VALUES (2,'Low','Low resolution image (roughly 500x500 pixels
or less)');
INSERT INTO resolutions VALUES (3,'Medium','Medium-resolution image (generally around
the 800-1000 pixel squared range).');
INSERT INTO resolutions VALUES (4,'High','High resolution image, larger than 1000
pixels squared.');

CREATE TABLE texts (
  id int(11) NOT NULL auto_increment,
  title text,
  author tinytext,
  publisher tinytext,
  city tinytext,
  doctype int(11) default NULL,
  date_num int(11) default NULL,
  date_text tinytext,
  notes text,
  pagination int(11) default NULL,
  illustrations text,
  category tinytext,
  dimensions text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;
```

Figure A-5



Streetprint: Revolution and Romanticism
http://localhost/streetprint-ver1/index.php
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
?

Welcome to Streetprint: Revolution and Romanticism.

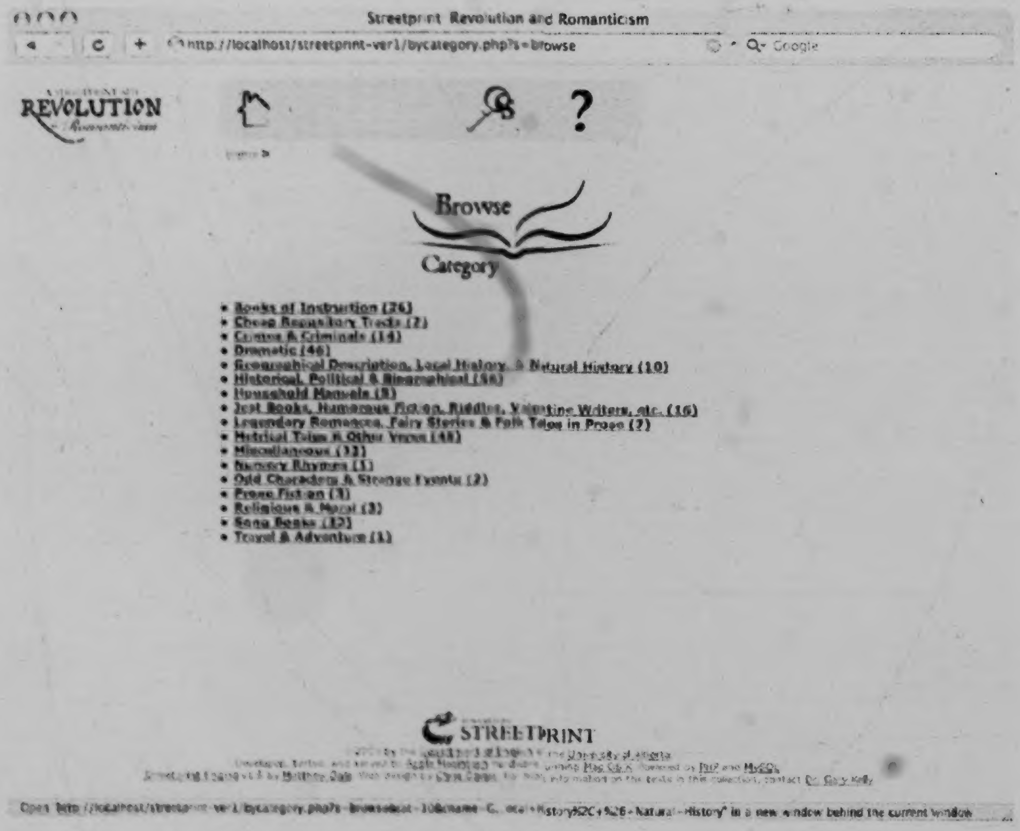
Streetprint: Revolution and Romanticism is a private collection of street literature held in Edmonton, Alberta, Canada. It comprises a wide range of types, from street ballads through chapbooks and tracts to valentines, from Britain and mostly from the late eighteenth and early nineteenth centuries. This website is powered by the Streetprint Engine, and can be easily browsed or searched in a variety of ways. The Streetprint engine is designed to accommodate with minimal alteration collections of other kinds of street literature.

Total texts in database: 183
Total fulltext records in database: 9
Total images in database: 1416
Total media files in database: 7

Featured text:

Anonymous, By varied arts how often has your Maid...
[illegible]

Figure A-6
Revolution and Romanticism, front page
September 2002

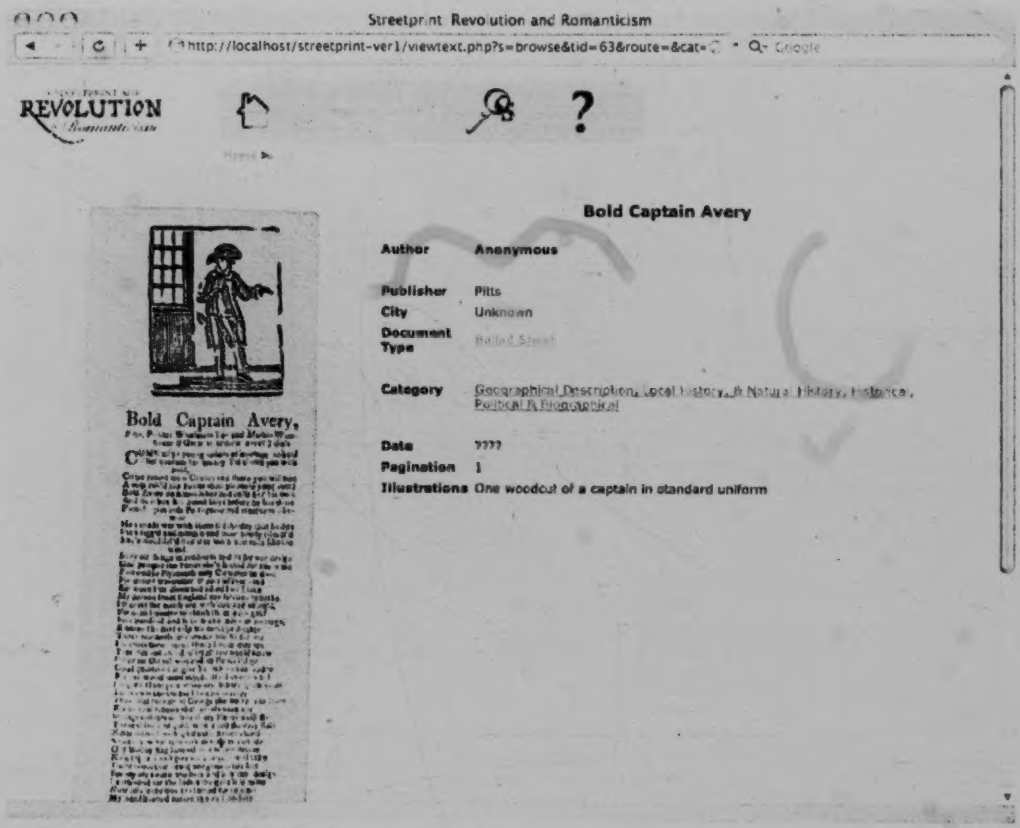

Streetprint Revolution and Romanticism
http://localhost/streetprint-ver1/bycategory.php?s=browse
Google
REVOLUTION
?
Browse
Category
STREETPRINT

- Books of Instruction (26)
- Cheap Repository Tracts (2)
- Crimes & Criminals (14)
- Dramatic (46)
- Geographical Description, Local History, & Natural History (10)
- Historical, Political & Biographical (56)
- Household Manuals (5)
- Jest Books, Humorous Fiction, Riddles, Valentine Writers, etc. (16)
- Legendary Romances, Fairy Stories & Folk Tales in Prose (7)
- Metrical Tales & Other Verse (48)
- Miscellaneous (32)
- Nursery Rhymes (1)
- Odd Characters & Strange Events (2)
- Prose Fiction (3)
- Religious & Moral (3)
- Song Books (12)
- Travel & Adventure (1)

Figure A-7

Revolution and Romanticism, browsing by category

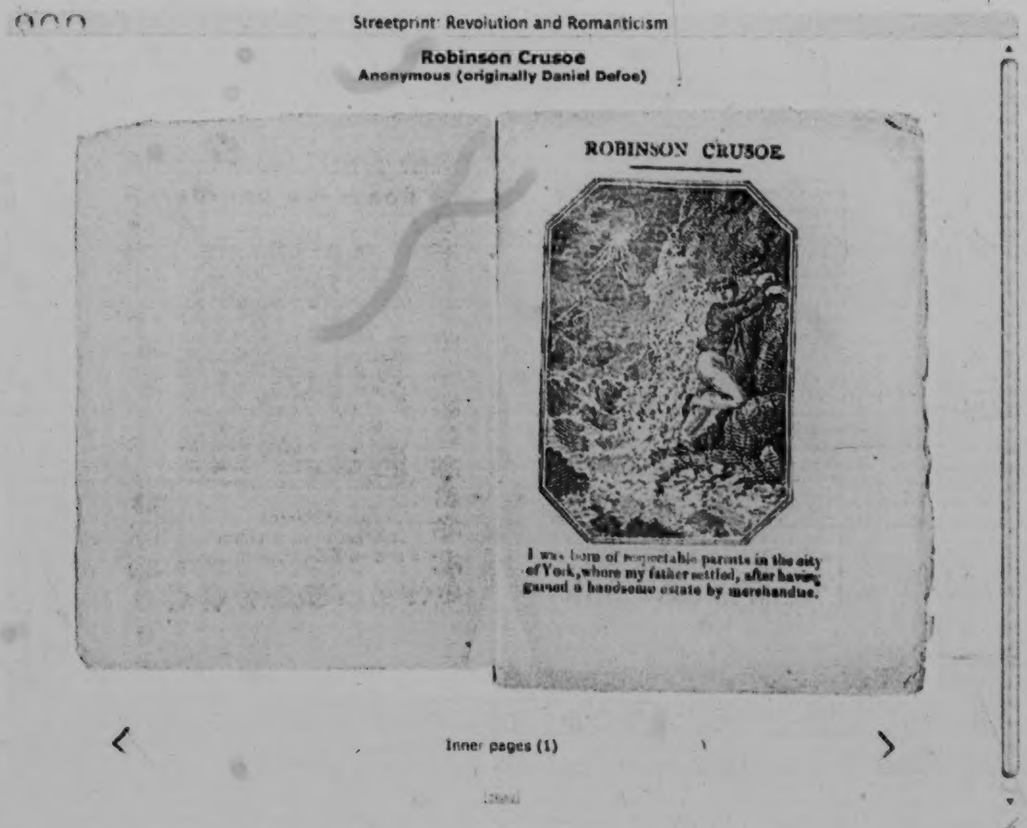
September 2002


Streetprint Revolution and Romanticism
http://localhost/streetprint-ver1/viewtext.php?s=browse&tid=63&route=&cat=
Google
REVOLUTION
Romanticism
Home

Bold Captain Avery

Author	**Anonymous**
Publisher	Pitts
City	Unknown
Document Type	Ballad Sheet
Category	Geographical Description, Local History, & Natural History, Historical, Political & Biographical
Date	????
Pagination	1
Illustrations	One woodcut of a captain in standard uniform

Figure A-8
Revolution and Romanticism, viewing a ballad sheet
September 2002



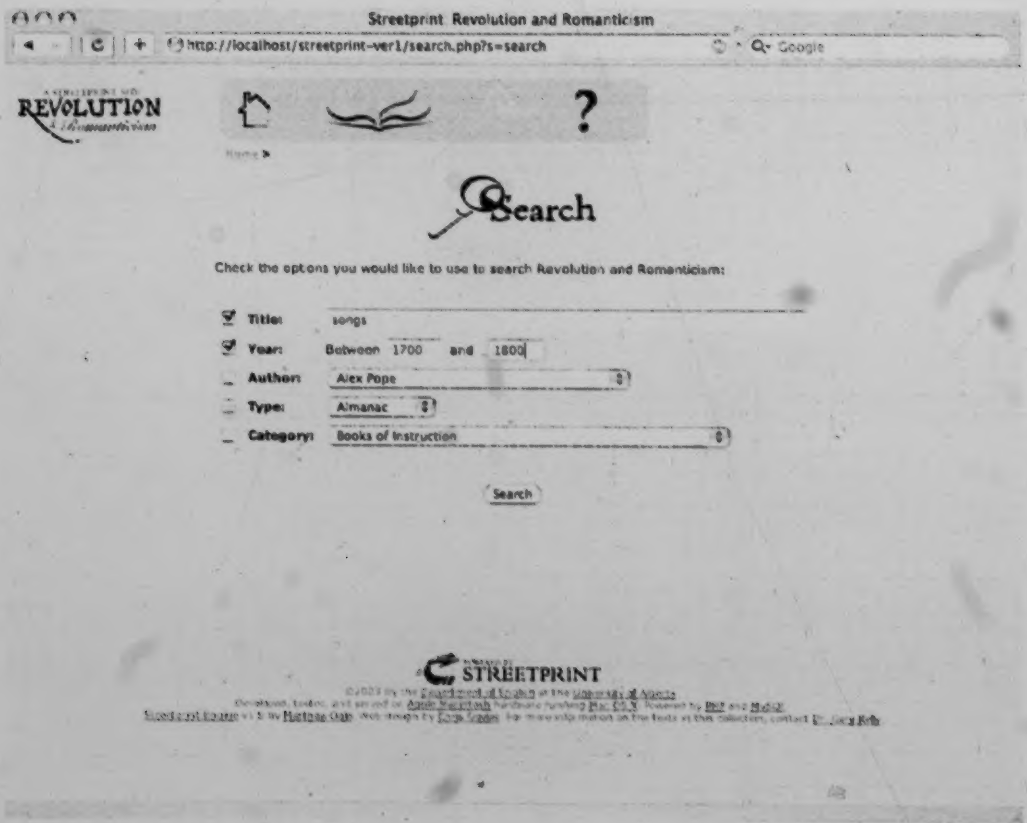
Streetprint: Revolution and Romanticism

Robinson Crusoe
Anonymous (originally Daniel Defoe)

ROBINSON CRUSOE.

I was born of respectable parents in the city of York, where my father settled, after having gained a handsome estate by merchandise.

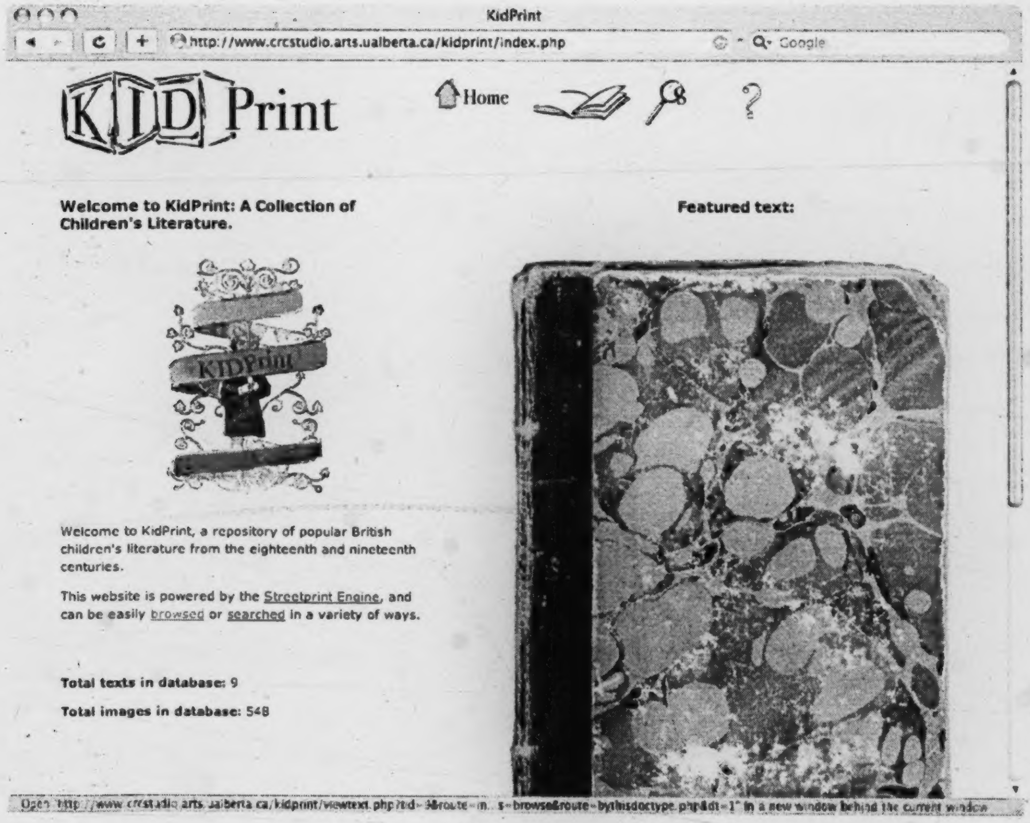
Figure A-9
Revolution and Romanticism, reading a chapbook
September 2002


Streetprint Revolution and Romanticism
http://localhost/streetprint-ver1/search.php?s=search
Google
REVOLUTION
?
Search
Search
STREETPRINT

Check the options you would like to use to search Revolution and Romanticism:

☑ **Title:** songs
☑ **Year:** Between 1700 and 1800
☐ **Author:** Alex Pope
☐ **Type:** Almanac
☐ **Category:** Books of Instruction

Figure A-10
Revolution and Romanticism, search screen
September 2002



KidPrint
http://www.crcstudio.arts.ualberta.ca/kidprint/index.php
Google
KID Print
Home
?

Welcome to KidPrint: A Collection of Children's Literature.

Featured text:

Welcome to KidPrint, a repository of popular British children's literature from the eighteenth and nineteenth centuries.

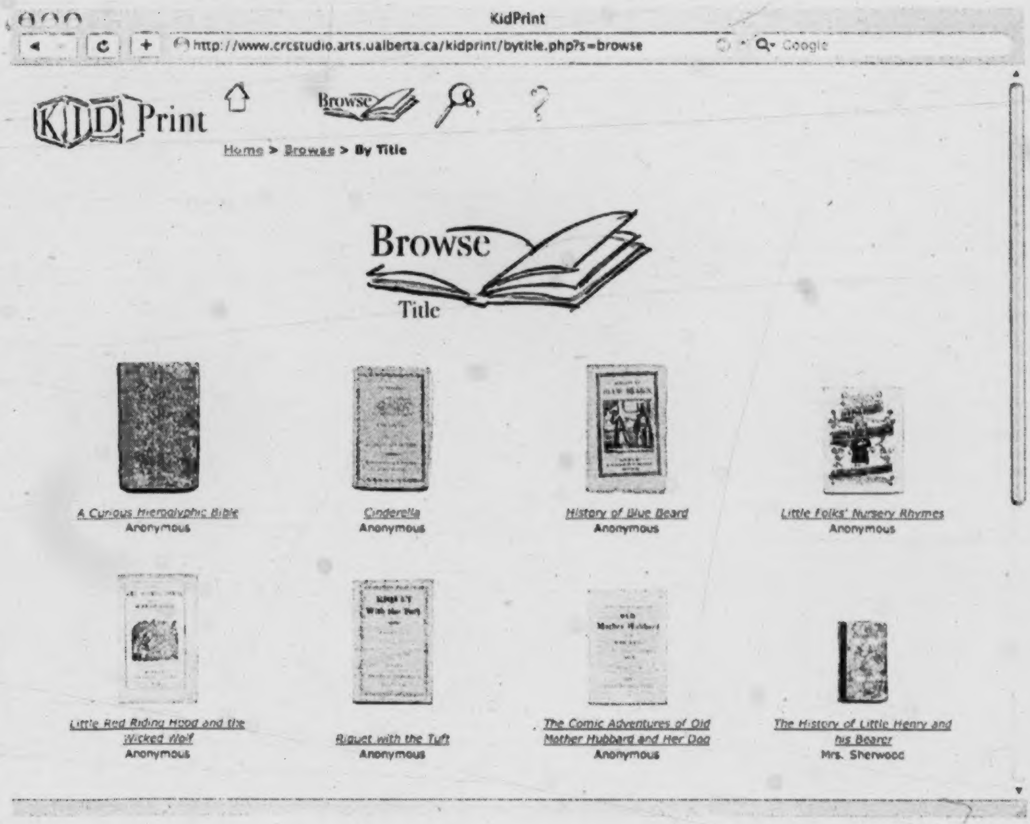
This website is powered by the Streetprint Engine, and can be easily browsed or searched in a variety of ways.

Total texts in database: 9

Total images in database: 548

Open "http://www.crcstudio.arts.ualberta.ca/kidprint/viewtext.php?tid=9&route=m...s=browse&route=bythisdoctype.php&dt=1" in a new window behind the current window

Figure A-11
Kidprint, front page
September 2002


KidPrint
http://www.crcstudio.arts.ualberta.ca/kidprint/bytitle.php?s=browse
Google
KID Print
Browse
Home > Browse > By Title
Browse
Title

A Curious Hieroglyphic Bible
Anonymous

Cinderella
Anonymous

History of Blue Beard
Anonymous

Little Folks' Nursery Rhymes
Anonymous

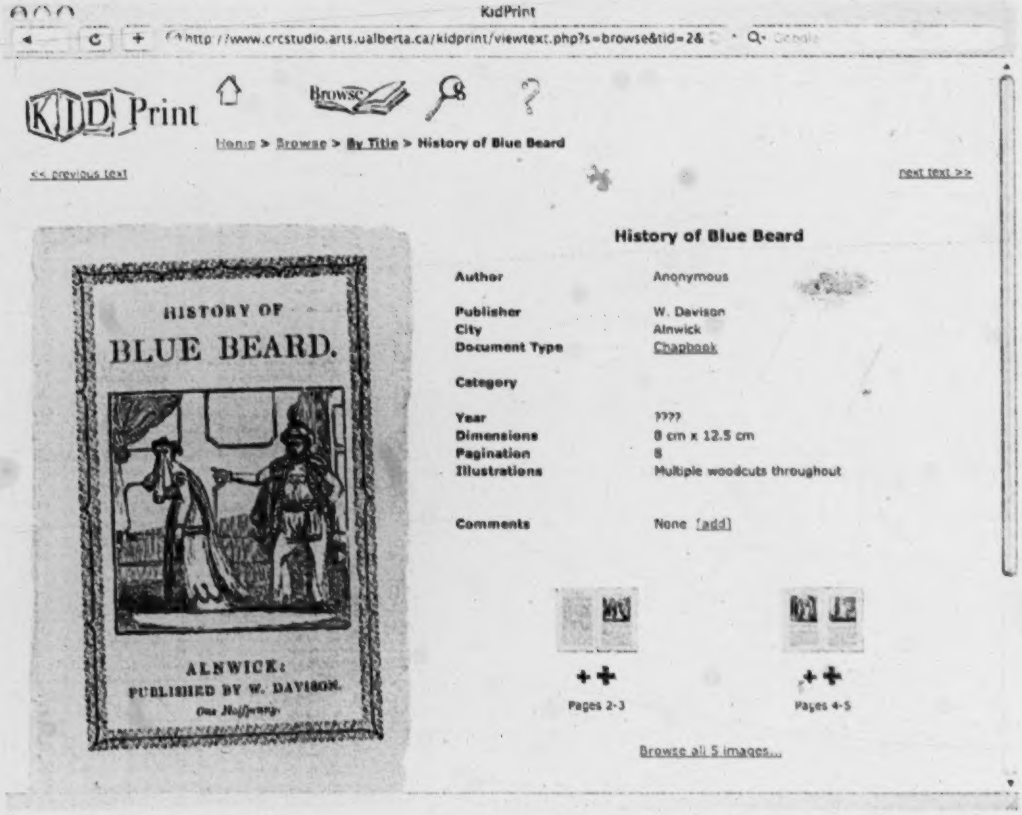
Little Red Riding Hood and the Wicked Wolf
Anonymous

Riquet with the Tuft
Anonymous

The Comic Adventures of Old Mother Hubbard and Her Dog
Anonymous

The History of Little Henry and his Bearer
Mrs. Sherwood

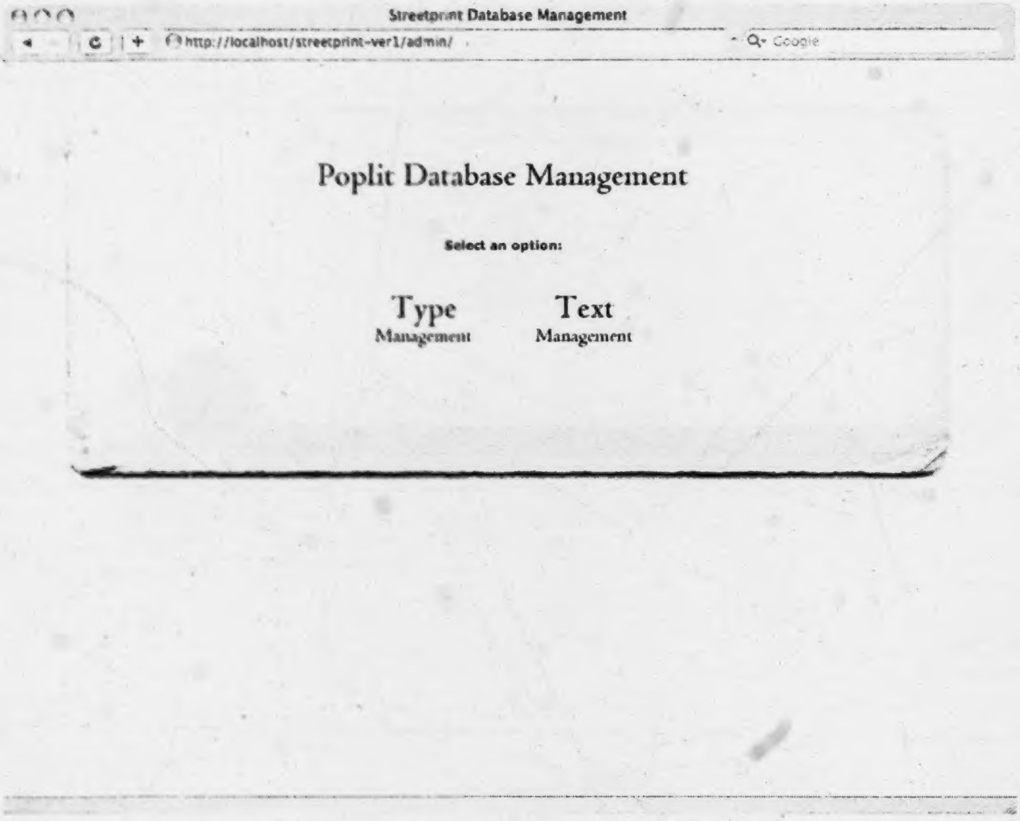
Figure A-12
Kidprint, browsing by title
September 2002


KidPrint
http://www.crcstudio.arts.ualberta.ca/kidprint/viewtext.php?s=browse&tid=2&
KID Print
Browse
Home > Browse > By Title > History of Blue Beard
<< previous text
next text >>
HISTORY OF
BLUE BEARD.
ALNWICK:
PUBLISHED BY W. DAVISON.
One Halfpenny.
Pages 2-3
Pages 4-5
Browse all 5 images...

History of Blue Beard

Author	Anonymous
Publisher	W. Davison
City	Alnwick
Document Type	Chapbook
Category	
Year	????
Dimensions	8 cm x 12.5 cm
Pagination	8
Illustrations	Multiple woodcuts throughout
Comments	None [add]

Figure A-13
Kidprint, viewing a text
September 2002


Streetprint Database Management
http://localhost/streetprint-ver1/admin/
Google

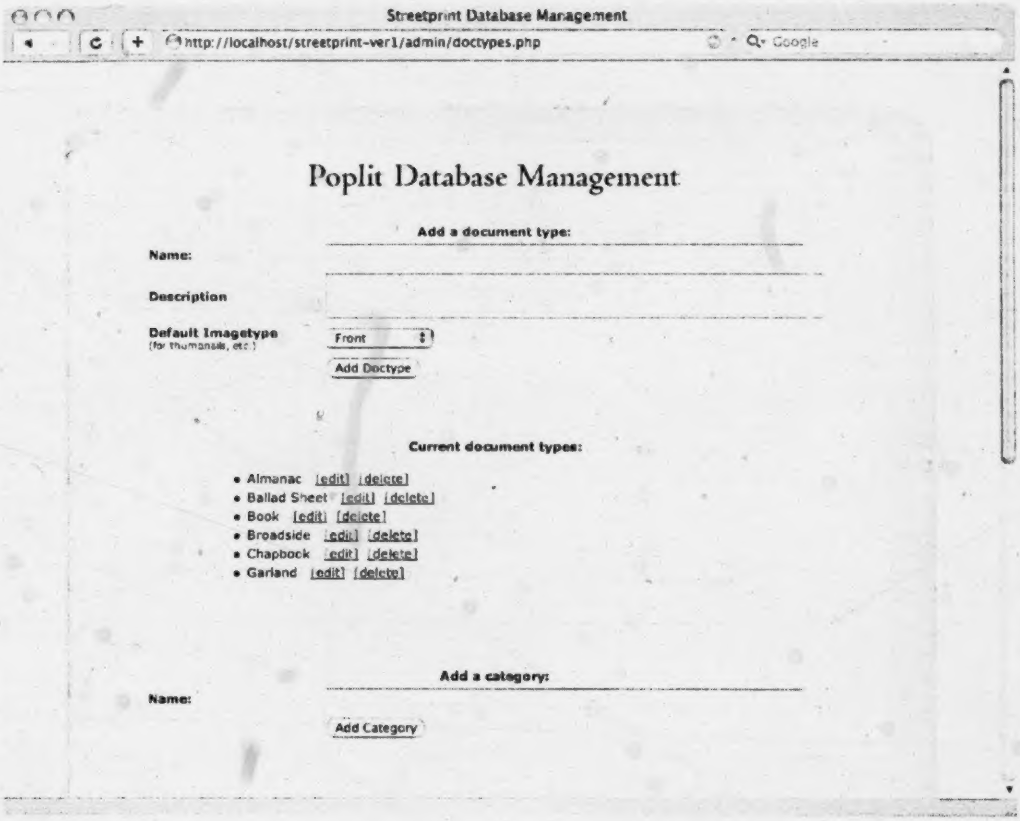
Poplit Database Management

Select an option:

Type Management

Text Management

Figure A-14
Streetprint 1.5 Administrative Tools
September 2002



Streetprint Database Management

http://localhost/streetprint-ver1/admin/doctypes.php

Poplit Database Management

Add a document type:

Name:

Description

Default Imagetype (for thumbnails, etc.): Front

Current document types:

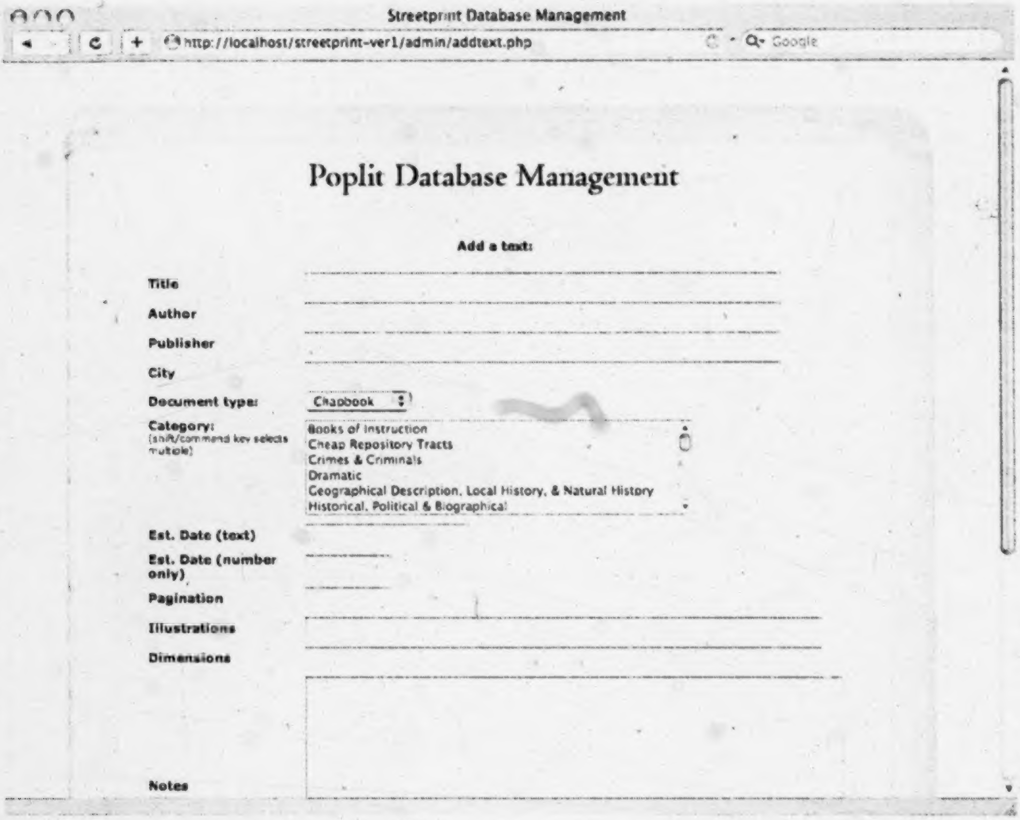
- Almanac [edit] [delete]
- Ballad Sheet [edit] [delete]
- Book [edit] [delete]
- Broadside [edit] [delete]
- Chapbook [edit] [delete]
- Garland [edit] [delete]

Add a category:

Name:

Figure A-15

Streetprint 1.5 Administrative Tools, document types
September 2002


Streetprint Database Management
http://localhost/streetprint-ver1/admin/addtext.php
Google

Poplit Database Management

Add a text:

Title

Author

Publisher

City

Document type: Chapbook

Category:
(shift/command key selects multiple)

Books of Instruction
Cheap Repository Tracts
Crimes & Criminals
Dramatic
Geographical Description, Local History, & Natural History
Historical, Political & Biographical

Est. Date (text)

Est. Date (number only)

Pagination

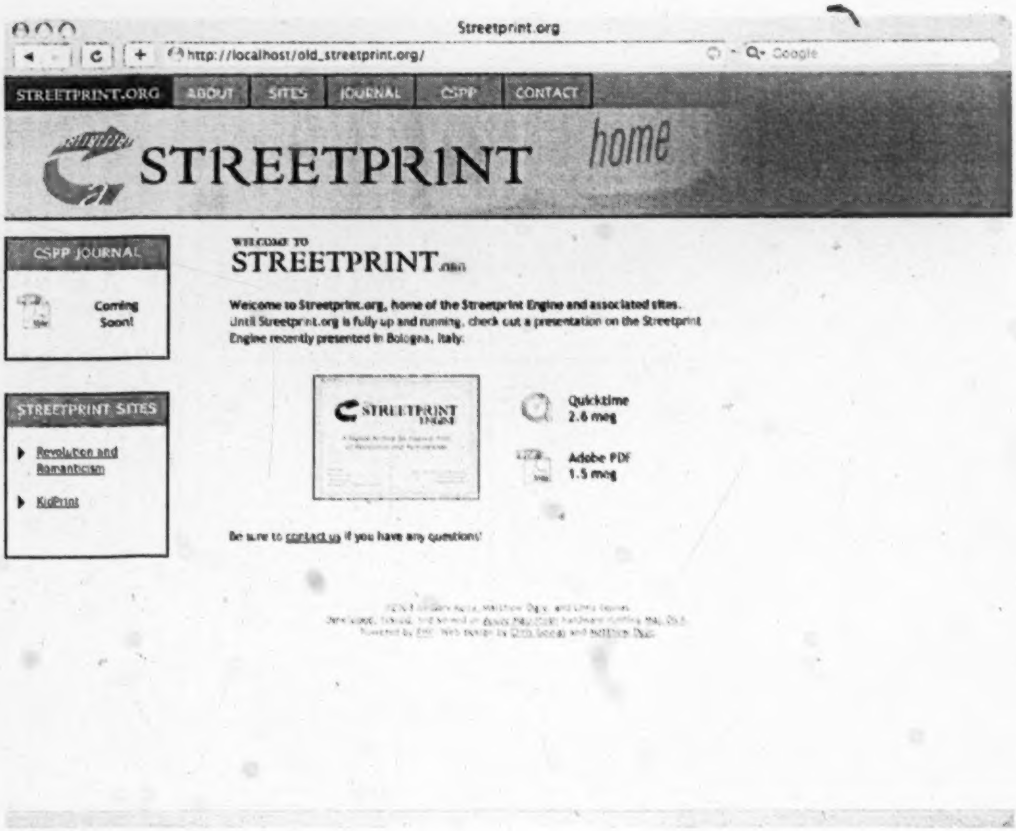
Illustrations

Dimensions

Notes

Figure A-16

Streetprint 1.5 Administrative Tools, adding a text
September 2002

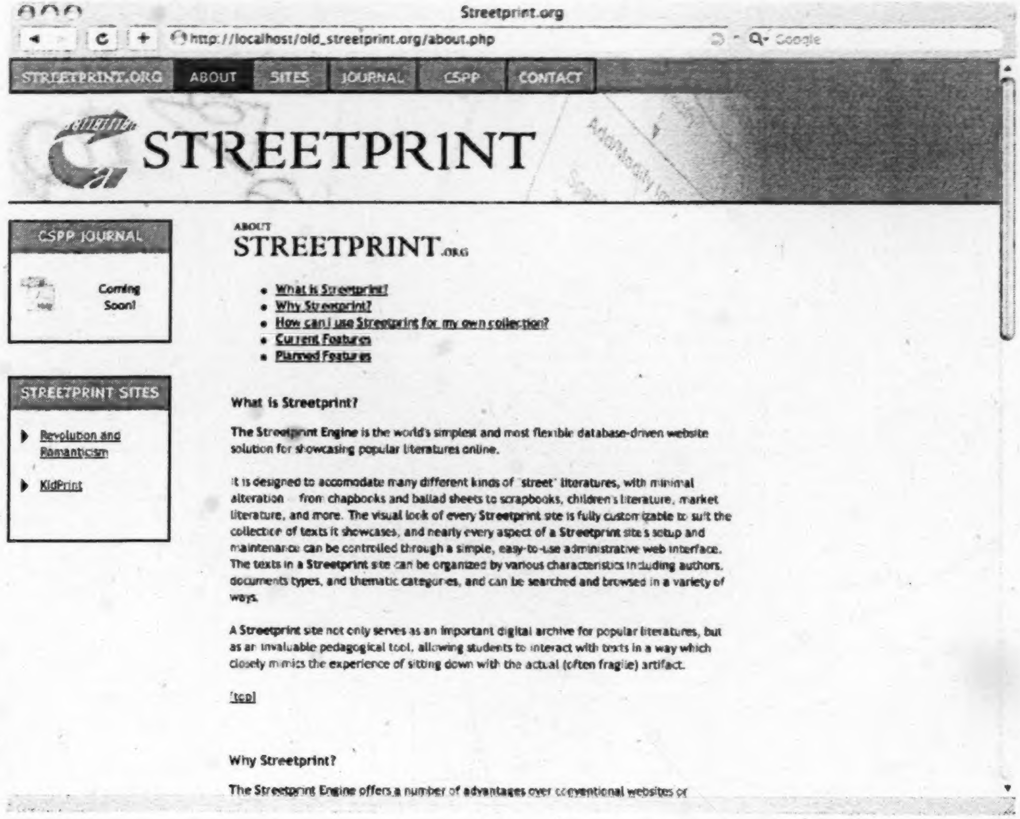

Streetprint.org
http://localhost/old_streetprint.org/
Google
STREETPRINT.ORG
ABOUT
SITES
JOURNAL
CSPP
CONTACT
STREETPRINT
home
CSPP JOURNAL
Coming Soon!
STREETPRINT SITES
Revolution and Romanticism
KidPrint
STREETPRINT ENGINE
Quicktime 2.6 meg
Adobe PDF 1.5 meg

WELCOME TO
STREETPRINT.ORG

Welcome to Streetprint.org, home of the Streetprint Engine and associated sites.
Until Streetprint.org is fully up and running, check out a presentation on the Streetprint Engine recently presented in Bologna, Italy.

Be sure to contact us if you have any questions!

Figure A-17
Streetprint.org, first version
January 2003


Streetprint.org
http://localhost/old_streetprint.org/about.php
Google
STREETPRINT.ORG
ABOUT
SITES
JOURNAL
CSPP
CONTACT
STREETPRINT
CSPP JOURNAL
Coming Soon!
STREETPRINT SITES
Revolution and Romanticism
KidPrint

ABOUT
STREETPRINT.ORG

- What is Streetprint?
- Why Streetprint?
- How can I use Streetprint for my own collection?
- Current Features
- Planned Features

What is Streetprint?

The Streetprint Engine is the world's simplest and most flexible database-driven website solution for showcasing popular literatures online.

It is designed to accomodate many different kinds of "street" literatures, with minimal alteration — from chapbooks and ballad sheets to scrapbooks, children's literature, market literature, and more. The visual look of every Streetprint site is fully customizable to suit the collection of texts it showcases, and nearly every aspect of a Streetprint site's setup and maintenance can be controlled through a simple, easy-to-use administrative web interface. The texts in a Streetprint site can be organized by various characteristics including authors, documents types, and thematic categories, and can be searched and browsed in a variety of ways.

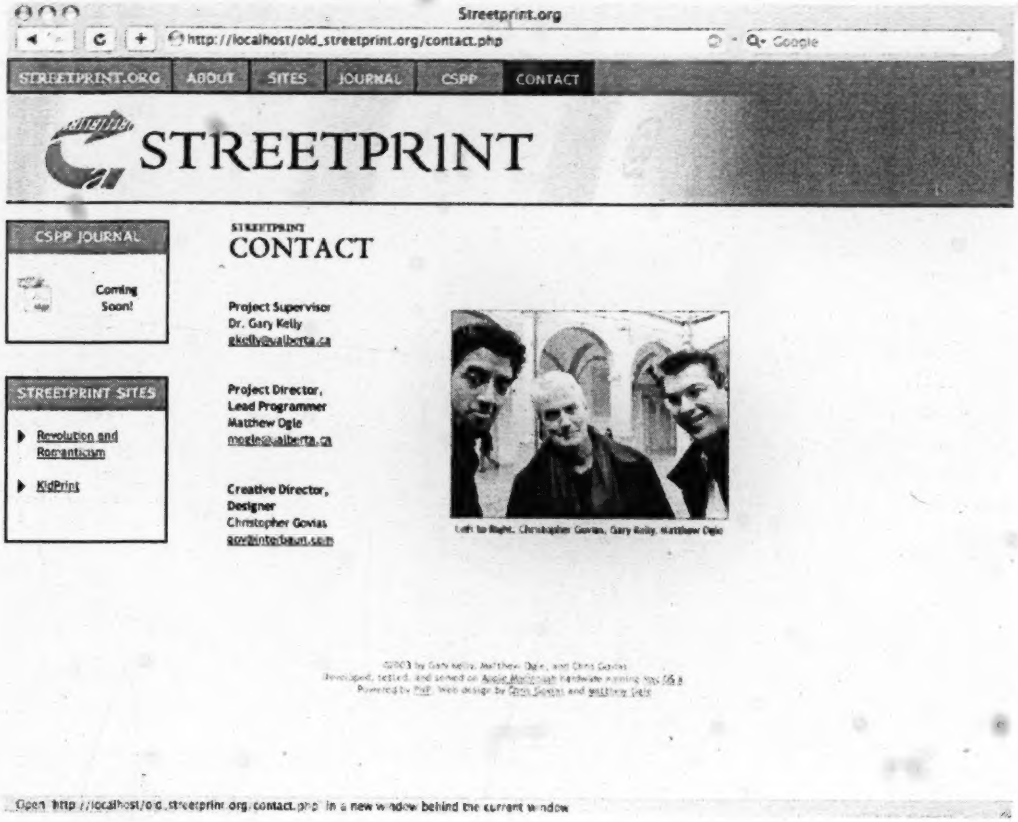
A Streetprint site not only serves as an important digital archive for popular literatures, but as an invaluable pedagogical tool, allowing students to interact with texts in a way which closely mimics the experience of sitting down with the actual (often fragile) artifact.

[top]

Why Streetprint?

The Streetprint Engine offers a number of advantages over conventional websites or

Figure A-18
Streetprint.org, About page
January 2003


Streetprint.org
http://localhost/old_streetprint.org/contact.php
Google
STREETPRINT.ORG
ABOUT
SITES
JOURNAL
CSPP
CONTACT
STREETPRINT
CSPP JOURNAL
Coming Soon!
STREETPRINT SITES
Revolution and Romanticism
KidPrint

STREETPRINT

CONTACT

Project Supervisor
Dr. Gary Kelly
gkelly@ualberta.ca

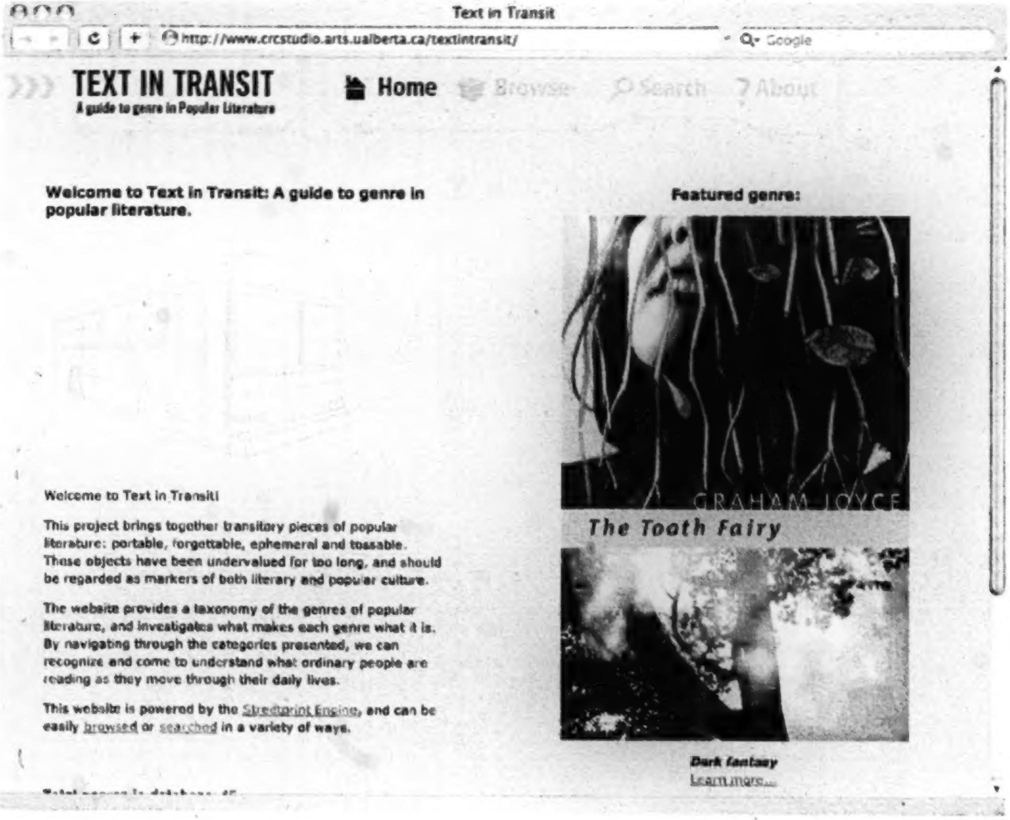
Project Director,
Lead Programmer
Matthew Ogle
mogle@ualberta.ca

Creative Director,
Designer
Christopher Govias
gov@interbaum.com

Left to Right: Christopher Govias, Gary Kelly, Matthew Ogle

Open "http://localhost/old_streetprint.org/contact.php" in a new window behind the current window

Figure A-19
Streetprint.org, Contact page
January 2003


Text in Transit
http://www.crcstudio.arts.ualberta.ca/textintransit/
Google
TEXT IN TRANSIT
A guide to genre in Popular Literature
Home
Browse
Search
About
GRAHAM JOYCE
The Tooth Fairy

Welcome to Text in Transit: A guide to genre in popular literature.

Featured genre:

Welcome to Text in Transit!

This project brings together transitory pieces of popular literature: portable, forgettable, ephemeral and tossable. These objects have been undervalued for too long, and should be regarded as markers of both literary and popular culture.

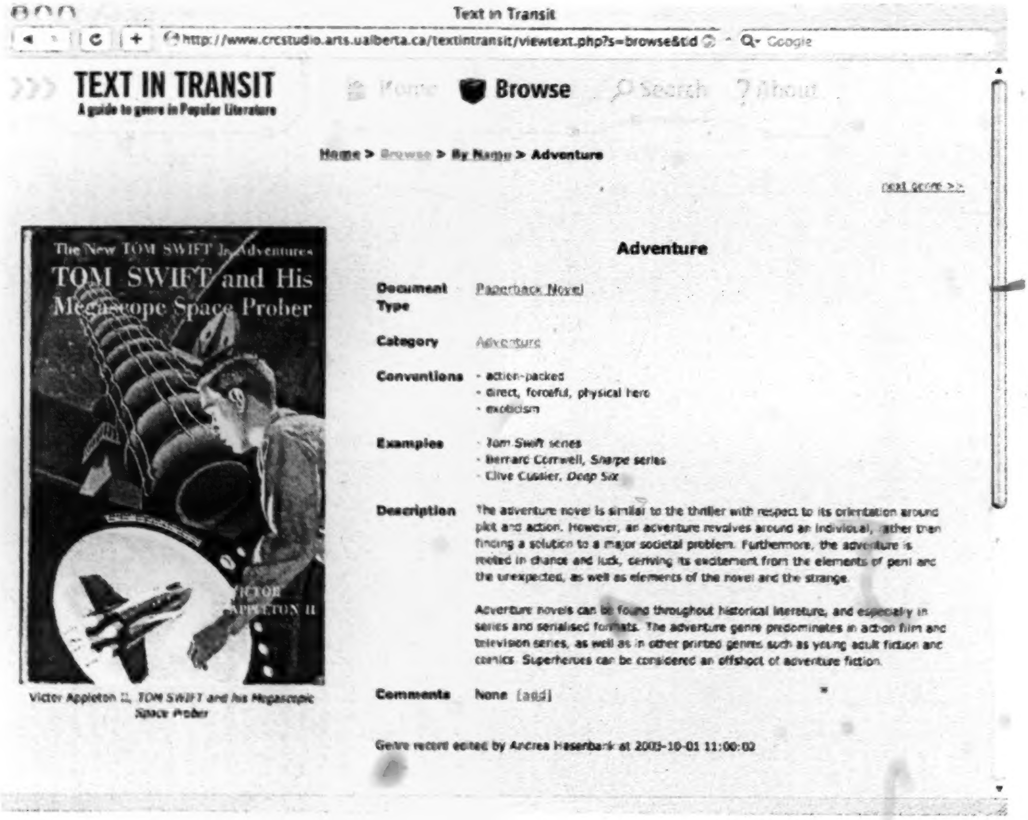
The website provides a taxonomy of the genres of popular literature, and investigates what makes each genre what it is. By navigating through the categories presented, we can recognize and come to understand what ordinary people are reading as they move through their daily lives.

This website is powered by the Streetprint Engine, and can be easily browsed or searched in a variety of ways.

Dark fantasy
Learn more...

Total genres in database: 45

Figure A-20
Text in Transit, front page


Text in Transit
http://www.crcstudio.arts.ualberta.ca/textintransit/viewtext.php?s=browse&tid
Google
TEXT IN TRANSIT
A guide to genre in Popular Literature
Home
Browse
Search
About
Home > Browse > By Name > Adventure
next genre >>
The New TOM SWIFT Jr. Adventures
TOM SWIFT and His Megascope Space Prober
VICTOR APPLETON II

Victor Appleton II, *TOM SWIFT and his Megascopic Space Prober*

Adventure

Document Type Paperback Novel

Category Adventure

Conventions
- action-packed
- direct, forceful, physical hero
- exoticism

Examples
- *Tom Swift* series
- Bernard Cornwell, *Sharpe* series
- Clive Cussler, *Deep Six*

Description The adventure novel is similar to the thriller with respect to its orientation around plot and action. However, an adventure revolves around an individual, rather than finding a solution to a major societal problem. Furthermore, the adventure is rooted in chance and luck, deriving its excitement from the elements of peril and the unexpected, as well as elements of the novel and the strange.

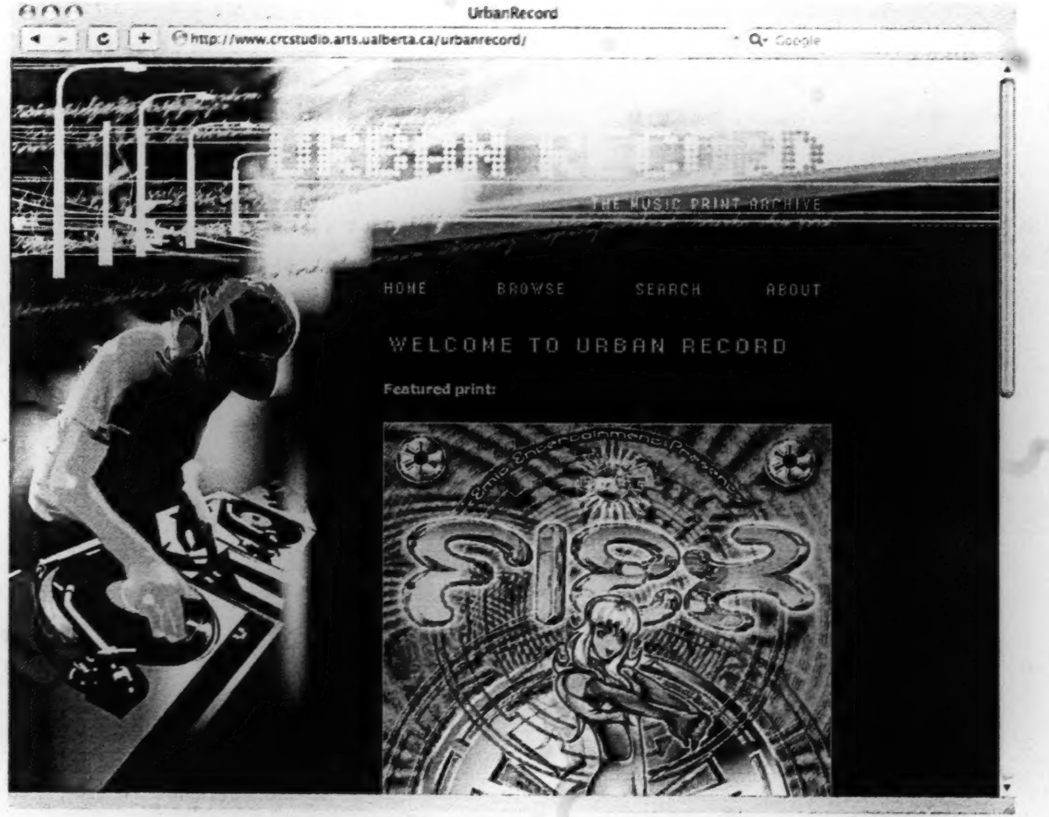
Adventure novels can be found throughout historical literature, and especially in series and serialised formats. The adventure genre predominates in action film and television series, as well as in other printed genres such as young adult fiction and comics. Superheroes can be considered an offshoot of adventure fiction.

Comments None [add]

Genre record edited by Andrea Hasenbank at 2005-10-01 11:00:02

Figure A-21

Text in Transit, viewing a genre


UrbanRecord
http://www.crcstudio.arts.ualberta.ca/urbanrecord/
Google
THE MUSIC PRINT ARCHIVE
HOME
BROWSE
SEARCH
ABOUT

WELCOME TO URBAN RECORD

Featured print:

Figure A-22
Urban Record, front page

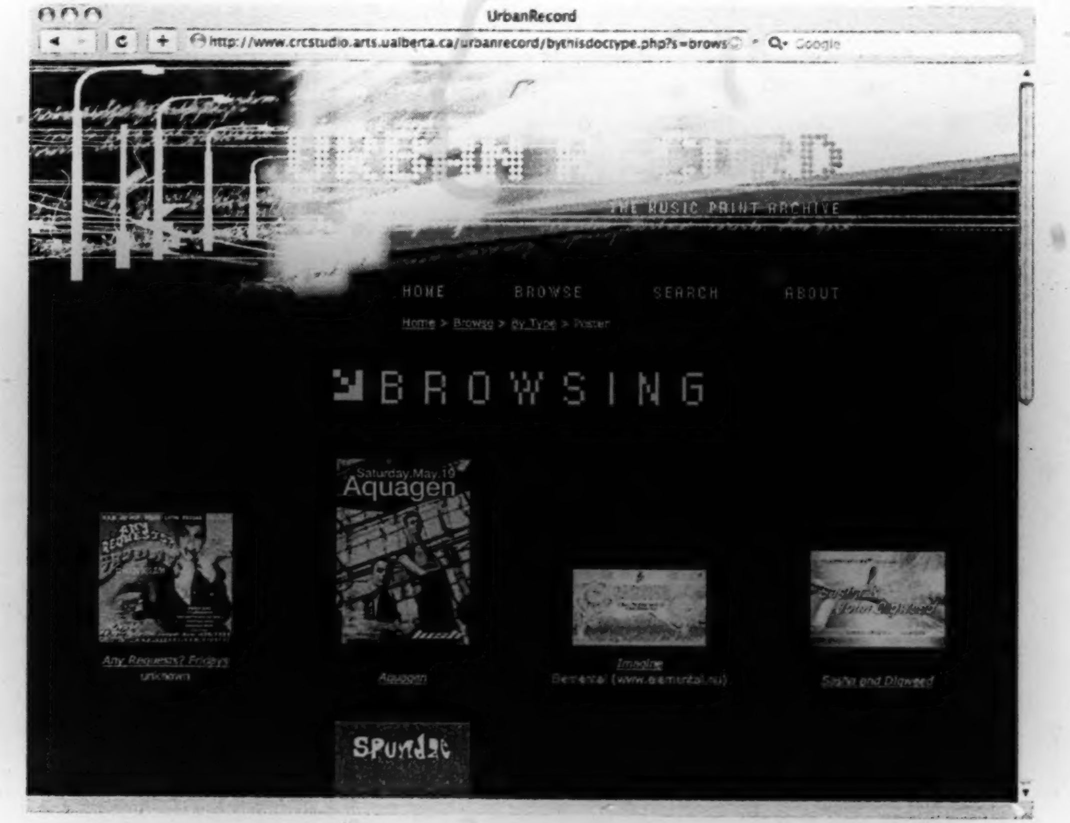
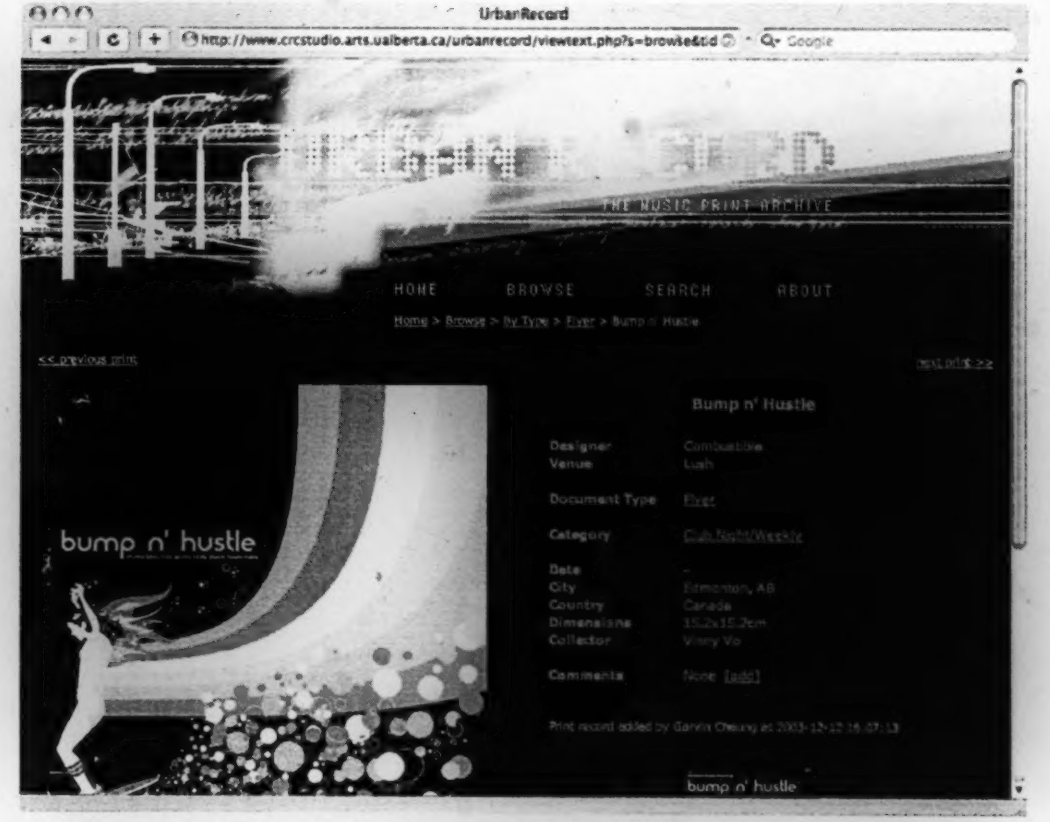

UrbanRecord
http://www.crcstudio.arts.ualberta.ca/urbanrecord/bythisdoctype.php?s=brows
Google
THE MUSIC PRINT ARCHIVE
HOME
BROWSE
SEARCH
ABOUT
Home > Browse > By Type > Poster
BROWSING
Saturday.May.19
Aquagen
Any Requests? Fridays
unknown
Aquagen
Imagine
Elemental (www.elemental.nu)
Sasha and Digweed
Spundae

Figure A-23
Urban Record, browsing posters


UrbanRecord
http://www.crcstudio.arts.ualberta.ca/urbanrecord/viewtext.php?s=browse&tid
Google
THE MUSIC PRINT ARCHIVE
HOME
BROWSE
SEARCH
ABOUT
Home > Browse > By Type > Flyer > Bump n' Hustle
<< previous print
next print >>
bump n' hustle

Bump n' Hustle

Designer Combustible
Venue Lush

Document Type Flyer

Category Club Night/Weekly

Date -
City Edmonton, AB
Country Canada
Dimensions 15.2x15.2cm
Collector Vinny Vo

Comments None [add]

Print record added by Garvin Cheung at 2003-12-12 16:07:13

Figure A-24
Urban Record, viewing a piece

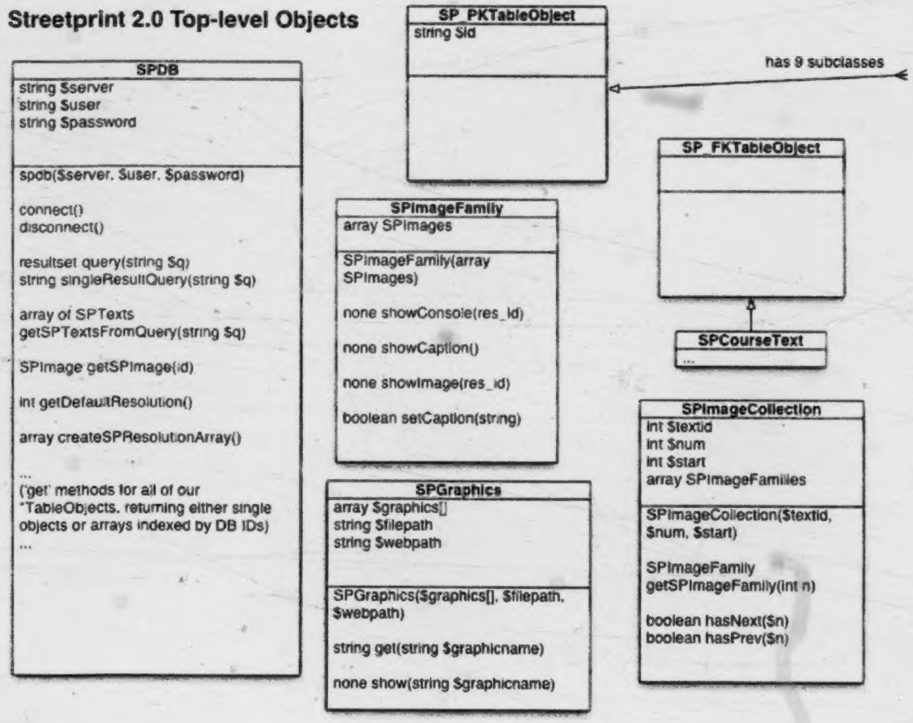

Streetprint 2.0 Top-level Objects
SPDB
string $server
string $user
string $password
spdb($server, $user, $password)
connect()
disconnect()
resultset query(string $q)
string singleResultQuery(string $q)
array of SPTexts
getSPTextsFromQuery(string $q)
SPImage getSPImage(id)
int getDefaultResolution()
array createSPResolutionArray()
...
('get' methods for all of our
*TableObjects, returning either single
objects or arrays indexed by DB IDs)
...
SP_PKTableObject
string $id
has 9 subclasses
SP_FKTableObject
SPCourseText
...
SPImageFamily
array SPImages
SPImageFamily(array
SPImages)
none showConsole(res_id)
none showCaption()
none showImage(res_id)
boolean setCaption(string)
SPGraphics
array $graphics[]
string $filepath
string $webpath
SPGraphics($graphics[], $filepath,
$webpath)
string get(string $graphicname)
none show(string $graphicname)
SPImageCollection
int $textid
int $num
int $start
array SPImageFamilies
SPImageCollection($textid,
$num, $start)
SPImageFamily
getSPImageFamily(int n)
boolean hasNext($n)
boolean hasPrev($n)

Figure A-25
Streetprint Engine 2.0 Object Planning

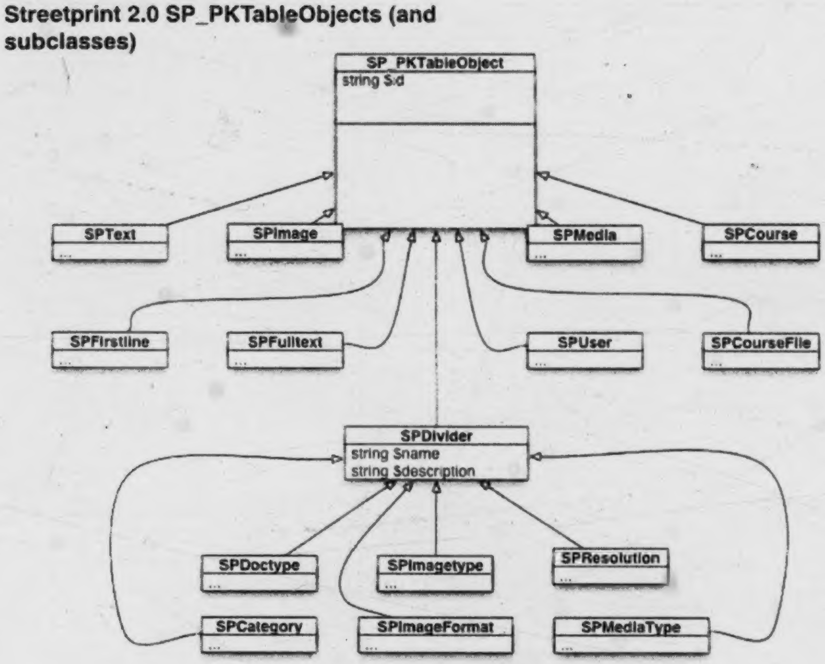
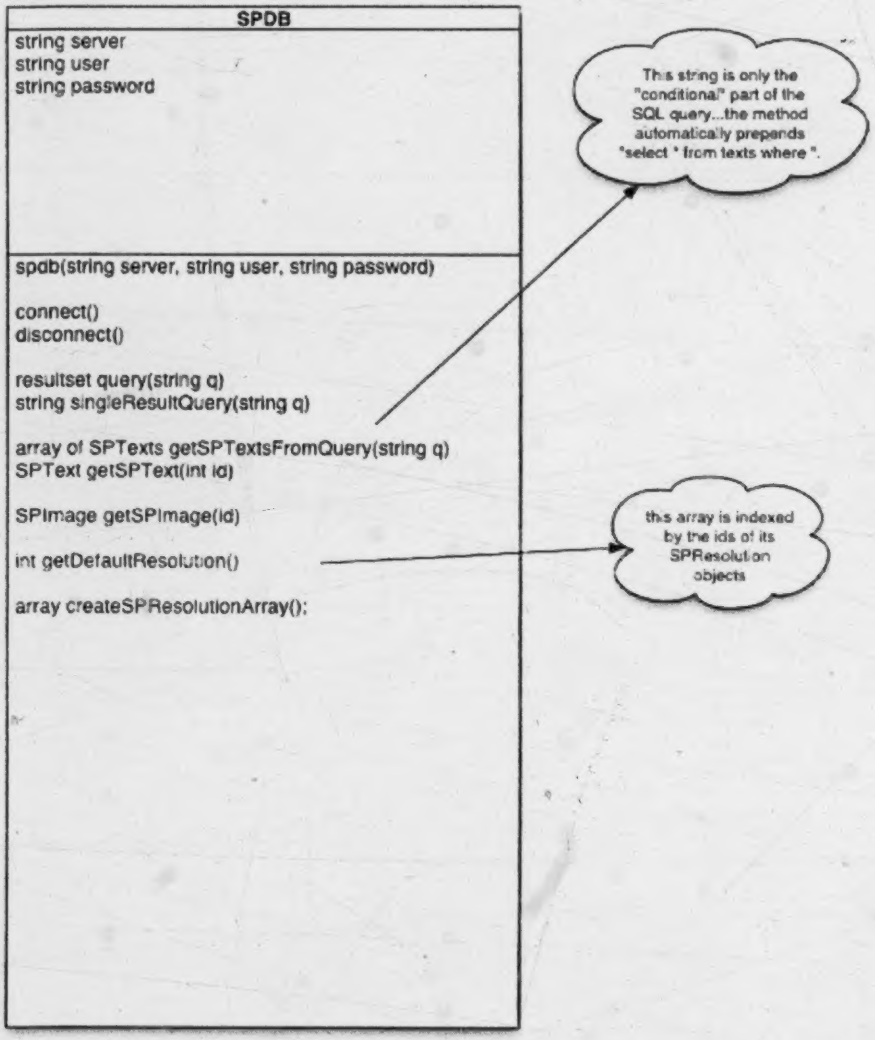

Streetprint 2.0 SP_PKTableObjects (and subclasses)
SP_PKTableObject
string $id
SPText
SPImage
SPMedia
SPCourse
SPFirstline
SPFulltext
SPUser
SPCourseFile
SPDivider
string $name
string $description
SPDoctype
SPImagetype
SPResolution
SPCategory
SPImageFormat
SPMediaType

Figure A-26
Streetprint Engine 2.0 Object Planning, cont.

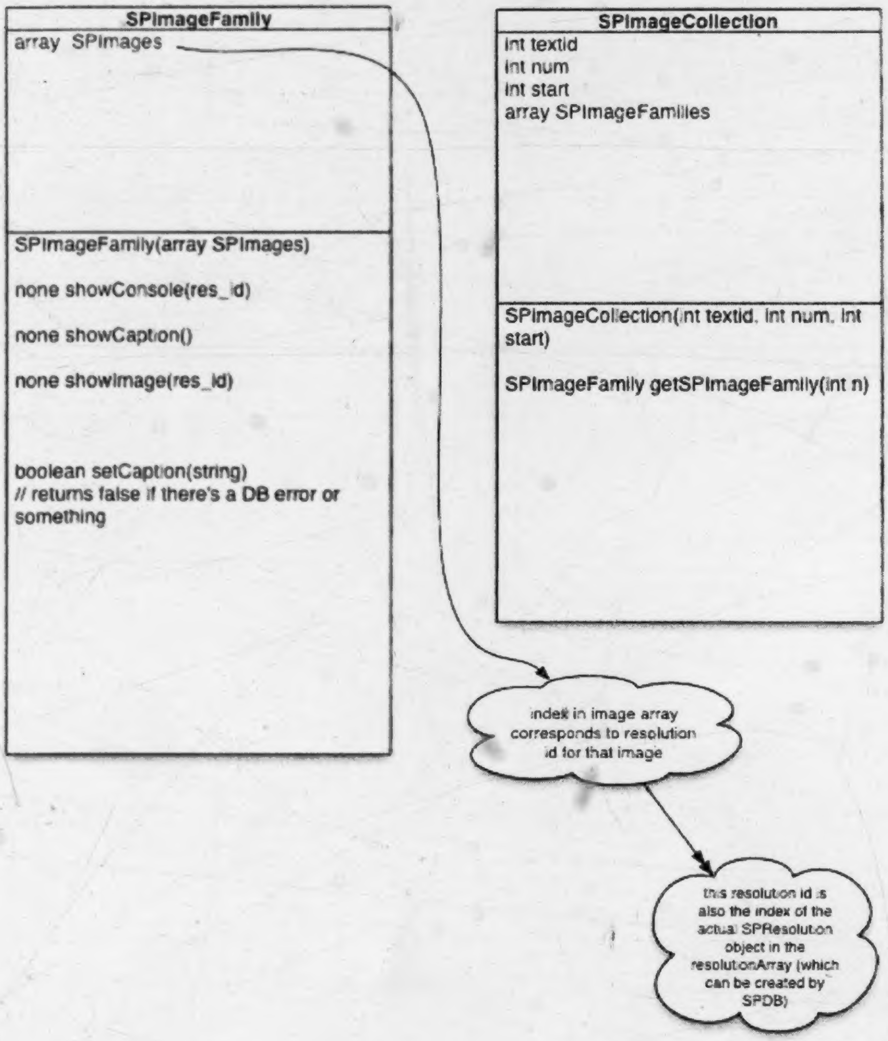


```
SPDB
string server
string user
string password

spdb(string server, string user, string password)

connect()
disconnect()

resultset query(string q)
string singleResultQuery(string q)

array of SPTexts getSPTextsFromQuery(string q)
SPText getSPText(int id)

SPImage getSPImage(id)

int getDefaultResolution()

array createSPResolutionArray();
```

This string is only the "conditional" part of the SQL query...the method automatically prepends "select * from texts where ".

this array is indexed by the ids of its SPResolution objects

Figure A-26
Streetprint Engine 2.0 Object Planning
SPDB Object


SPImageFamily
array SPImages
SPImageFamily(array SPImages)
none showConsole(res_id)
none showCaption()
none showImage(res_id)
boolean setCaption(string)
// returns false if there's a DB error or something
SPImageCollection
int textid
int num
int start
array SPImageFamilies
SPImageCollection(int textid, int num, int start)
SPImageFamily getSPImageFamily(int n)

index in image array corresponds to resolution id for that image

this resolution id is also the index of the actual SPResolution object in the resolutionArray (which can be created by SPDB)

Figure A-27
Streetprint Engine 2.0 Object Planning
SP Image Objects

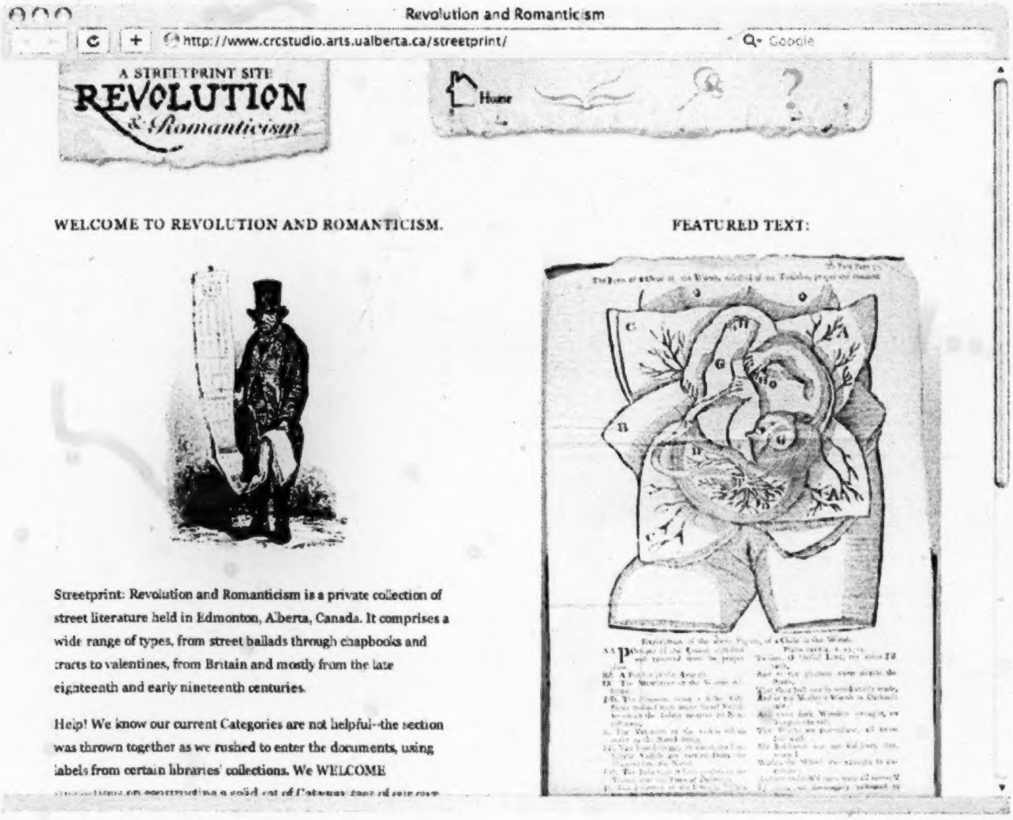
```
# STREETPRINT 2.0
# DATABASE SPEC
# 11 July 2003


# To build a new Streetprint MySQL database from this file, use this command:
#
#  mysql -u [user] -p [newdbname] < streetprint.sql
#
# (you must have already created your [newdbname] database and it should be empty)

# --------------------------------------------------------------------------------

#
# -----
# TEXTS
# -----
# and related tables
#

#
# Table structure for table 'texts'
#
# "category" contains a PHP serialized array of
# IDs from the categories table

CREATE TABLE texts (
  id int(11) NOT NULL auto_increment,
  title text,
  author text,
  publisher text,
  city text,
  doctype int(11) default NULL,
  date_num int(11) default NULL,
  date_text tinytext,
  notes text,
  pagination int(11) default NULL,
  illustrations text,
  category tinytext,
  dimensions text,
  location text,
  publish int,
  PRIMARY KEY  (id),
  FOREIGN KEY (doctype) REFERENCES doctypes(id)
) TYPE=MyISAM;


#
# Table structure for table 'categories'
# Categories are thematic divisions for texts, ie.
# Dramatic Verse, Humour, Non-fiction, etc.
#

CREATE TABLE categories (
  id int(11) NOT NULL auto_increment,
  name text,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

#
# Table structure for table 'doctypes'
# Doctypes are physical divisions for texts, ie.
# Broadside, chapbook, almanac, book, brochure, etc.
#
# def_imagetype = "default imagetype", so it knows
# what 'kind' of thumbnail to grab to represent this doctype

CREATE TABLE doctypes (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  def_imagetype int(11) default NULL,
  PRIMARY KEY  (id),
  FOREIGN KEY (def_imagetype) REFERENCES imagetypes(id)
) TYPE=MyISAM;

#
# Dumping data for table 'doctypes'
#

INSERT INTO doctypes VALUES (18,'Default','Default document type - change me to suit
your collection!', '1');


#
# Table structure for table 'firstlines'
# Stores first lines of poems contained in texts.
#

CREATE TABLE firstlines (
  id int(11) NOT NULL auto_increment,
  text_id int(11) default NULL,
  firstline text,
  PRIMARY KEY  (id),
  FOREIGN KEY (text_id) REFERENCES texts(id)
) TYPE=MyISAM;

#
# Table structure for table 'fulltexts'
# Stores the complete fulltext for a given text
#

CREATE TABLE fulltexts (
  id int(11) NOT NULL auto_increment,
  text_id int(11) NOT NULL default '0',
  full mediumtext,
  PRIMARY KEY  (id),
  FOREIGN KEY (text_id) REFERENCES texts(id)
) TYPE=MyISAM;


# --------------------------------------------------------------------------------
```

```
#
# -----
# IMAGES
# -----
# and related tables
#

#
# Table structure for table 'images'
#
# firstpagenum -> the first numbered page on this images (ie. "34")
# numpages -> total number of pages represented on this image (ie. "2")

CREATE TABLE images (
  id int(11) NOT NULL auto_increment,
  text_id int(11) default NULL,
  imagetype int(11) default NULL,
  resolution int(11) default NULL,
  caption text,
  format int(11) default NULL,
  firstpagenum int(11) default NULL,
  numpages int(11) default NULL,
  PRIMARY KEY  (id),
  FOREIGN KEY (text_id) REFERENCES texts(id),
  FOREIGN KEY (imagetype) REFERENCES imagetypes(id),
  FOREIGN KEY (resolution) REFERENCES revolutions(id),
  FOREIGN KEY (format) REFERENCES imageformats(id)
) TYPE=MyISAM;

#
# Table structure for table 'imagetypes'
# Kinds of images, ie. front cover, back cover, inside page,
# illustration, etc.
#

CREATE TABLE imagetypes (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

#
# Dumping data for table 'imagetypes'
#

INSERT INTO imagetypes VALUES (1,'Front','Front cover.');
INSERT INTO imagetypes VALUES (2,'Back','Back cover.');
INSERT INTO imagetypes VALUES (3,'Inside','Inner page.');

#
# Table structure for table 'imageformats'
# File formats for images. JPEG and GIF come pre-loaded.
#

CREATE TABLE imageformats (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  extension tinytext,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

#
# Dumping data for table 'imageformats'
#

INSERT INTO imageformats VALUES (1,'JPEG','JPEG Compression, best for
photographs.','jpg');
INSERT INTO imageformats VALUES (2,'GIF','GIF compression, best for line art/vector
art.','gif');
INSERT INTO imageformats VALUES (3,'PNG','PNG compression, lovely alpha
transparency.','png');

#
# Table structure for table 'resolutions'
# Different sizes for images. Some are included below.
#

CREATE TABLE resolutions (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

#
# Dumping data for table 'resolutions'
#

INSERT INTO resolutions VALUES (1,'Thumbnail','A small, minaturized representation of
an image.');
INSERT INTO resolutions VALUES (2,'Low','Low resolution image (roughly 500x500 pixels
or less)');
INSERT INTO resolutions VALUES (3,'Medium','Medium-resolution image (generally around
the 800-1000 pixel squared range).');
INSERT INTO resolutions VALUES (4,'High','High resolution image, larger than 1000
pixels squared.');


# --------------------------------------------------------------------------------

#
# -----
# MEDIA
# -----
# and related tables
#

#
# Table structure for table 'media'
# For any media associated with a text (songs, movies, etc.)
#

CREATE TABLE media (
  id int(11) NOT NULL auto_increment,
  text_id int(11) NOT NULL default '0',
  mediatype int(11) NOT NULL default '0',
  name tinytext,
  description text,
  PRIMARY KEY  (id),
  FOREIGN KEY (text_id) REFERENCES texts(id)
) TYPE=MyISAM;

#
# Table structure for table 'mediatypes'
# Types of media files, ie Quicktime, mp3, etc.
# 'Extension' is for the three-letter filename extension associated with
#  this media type.
#
CREATE TABLE mediatypes (
  id int(11) NOT NULL auto_increment,
  name tinytext,
  extension tinytext,
  description text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

#
# Dumping data for table 'mediatypes'
#

INSERT INTO mediatypes VALUES (1,'MP3 Audio File','Standard music/audio compression
format.','mp3');
INSERT INTO mediatypes VALUES (2,'Quicktime Movie','Apple\'s Quicktime video
format.','mov');

# --------------------------------------------------------------------------------

#
# -----
# COURSES
# -----
# and related tables
#



#
# Table structure for table 'courses'
# This table holds info on courses which will be taught
# using Streetprint syllabi.
#
# coursesubject -> ie. "ENGL"
# coursenum -> ie. "383"
# coursename -> "English 383"
# coursetopic -> "History of Literatures of Popular Culture"
# organize_by is a keyword specifying how to organize the
# course texts on the syllabus, ie. "bycategory", "bydoctype",
# "chronological", etc.
# when advertisecourse is 1, link appears on front page to this syllabus

CREATE TABLE courses (
  id int(11) NOT NULL auto_increment,
  coursesubject text,
  coursenum tinytext,
  coursename text,
  coursetopic text,
  coursedescription text,
  courseprof text,
  coursehours text,
  courseofficehours text,
  courseroom text,
  organizeby tinytext,
  advertisecourse int(11),
  PRIMARY KEY  (id)
) TYPE=MyISAM;

#
# Table structure for table 'coursetexts'
# This table holds info on texts associated with a
# specific course.
# Most texts will likely be in the Streetprint database;
# such records only require the courseid and textid
# fields. To specify a text which is not in the database
# (it will be listed on the syllabus without a hyperlink),
# leave the textid field blank and use the other fields as necessary.

CREATE TABLE coursetexts (
  id int(11) NOT NULL auto_increment,
  courseid int(11),
  textid int(11),
  textname text,
  textauthor text,
  PRIMARY KEY (id),
  FOREIGN KEY (courseid) REFERENCES courses(id),
  FOREIGN KEY (textid) REFERENCES texts(id)
) TYPE=MyISAM;

#
# Table structure for table 'coursefiles'
# Table for user-uploaded course-related files
# (PDFs, DOCs, PPTs, etc.)

CREATE TABLE coursefiles (
  id int(11) NOT NULL auto_increment,
  courseid int(11),
  filename text,
  filedesc text,
  PRIMARY KEY (id),
  FOREIGN KEY (courseid) REFERENCES courses(id)
) TYPE=MyISAM;


# --------------------------------------------------------------------------------

#
# -----
# USERS
# -----
# and related tables
#


#
# Table structure for table 'users'
# The default, blank, "admin" user is also created automatically
# 'roles' at the moment should be "editor" or "dataentry".
# Passwords are hashed with md5() before they enter the database.

CREATE TABLE users (
  id int(11) NOT NULL auto_increment,
  name text,
  password text,
  initials tinytext,
  email text,
  role tinytext,
  bio text,
  PRIMARY KEY(id)
) TYPE=MyISAM;

#
# Dumping data for table 'users'
#

INSERT INTO users VALUES (1,'Streetprint
Admin','','admin','admin@streetprint.org','editor','Default Streetprint user. Change
this info at your leisure.');


#
# Table structure for table 'archive_timestamps'
# 'action' can be "add", "edit", "delete"

CREATE TABLE timestamps (
  id int(11) NOT NULL auto_increment,
  date_time datetime NOT NULL,
  userid int(11),
  textid int(11),
  action tinytext,
  notes text,
  PRIMARY KEY (id),
  FOREIGN KEY (userid) REFERENCES users(id),
  FOREIGN KEY (textid) REFERENCES texts(id)
) TYPE=MyISAM;


# --------------------------------------------------------------------------------

#
# -----
# SITE CONFIG
# -----
#


#
# Table structure for table 'spconfig'
# This is the Streetprint site configuration table.
# It contains only one record, which is automatically entered
# below. It stores site-related information.
#
# pagetitle appears in the browser's title bar + on index page, ie. "KidPrint"
# sitename is used to refer to the site on various sub-pages,
# ie. "KidPrint: A Collection of Children's Literature".
# welcomeblurb appears as intro text on the site's front page
# browseby_.... and searchby_.... are options to specify what kind of
# browsing/searching is permitted. 1 turns them on, 0 turns them off

CREATE TABLE spconfig (
    pagetitle text,
    sitename text,
    welcomeblurb text,
    defaultresolution int(11) default NULL,
    featuredtext int(11) default NULL,
    featuredtext_img int(11) default NULL,
    browse_preferences text,
    search_preferences text,
    aboutproject text,
    aboutprocedures text,
    textsingular tinytext,
    textplural tinytext,
    fineprint text,
    additionalconfig text,
    displayuserinfo tinytext,
    FOREIGN KEY (defaultresolution) REFERENCES resolutions(id),
    FOREIGN KEY (featuredtext) REFERENCES texts(id),
    FOREIGN KEY (featuredtext_img) REFERENCES images(id)
)  TYPE=MyISAM;

#
# Dumping data for table 'spconfig'
# Adds the default config settings
#

INSERT INTO spconfig VALUES ('Streetprint', 'A Streetprint Site', 'Welcome to A
Streetprint Site. (Put your welcome text here!)', '2', '', '', '', '', 'Put
paragraphs describing your streetprint site here.', 'Put text describing your
archiving practices and methodologies here.', 'text', 'texts', '','', 'false');
```

Figure A-28

Streetprint Engine 2.0 Database Specification

July 2003


Revolution and Romanticism
http://www.crcstudio.arts.ualberta.ca/streetprint/
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
Home
?

WELCOME TO REVOLUTION AND ROMANTICISM.

FEATURED TEXT:

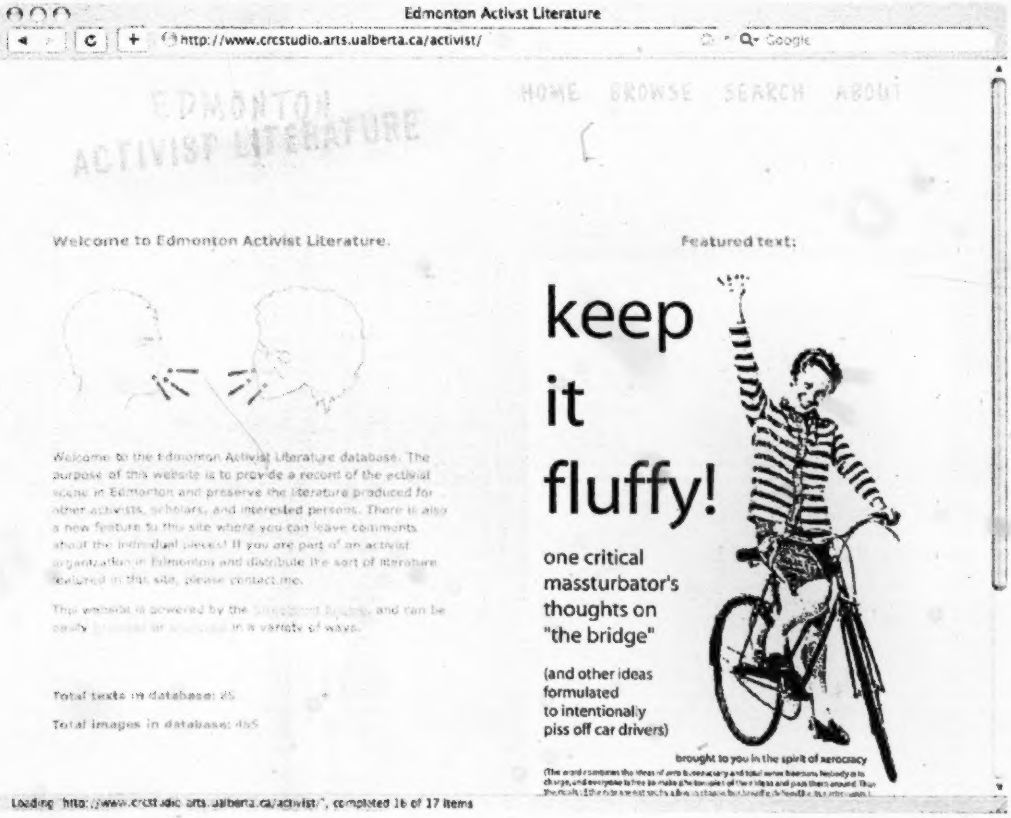
Streetprint: Revolution and Romanticism is a private collection of street literature held in Edmonton, Alberta, Canada. It comprises a wide range of types, from street ballads through chapbooks and tracts to valentines, from Britain and mostly from the late eighteenth and early nineteenth centuries.

Help! We know our current Categories are not helpful--the section was thrown together as we rushed to enter the documents, using labels from certain libraries' collections. We WELCOME

Figure A-29

Revolution and Romanticism 2.0

November 2003


Edmonton Activst Literature
http://www.crcstudio.arts.ualberta.ca/activist/
Google
EDMONTON ACTIVIST LITERATURE
HOME
BROWSE
SEARCH
ABOUT
Featured text:
keep
it
fluffy!
one critical
massturbator's
thoughts on
"the bridge"
(and other ideas
formulated
to intentionally
piss off car drivers)
brought to you in the spirit of aerocracy

Welcome to Edmonton Activist Literature.

Welcome to the Edmonton Activist Literature database. The purpose of this website is to provide a record of the activist scene in Edmonton and preserve the literature produced for other activists, scholars, and interested persons. There is also a new feature to this site where you can leave comments about the individual pieces! If you are part of an activist organization in Edmonton and distribute the sort of literature featured in this site, please contact me.

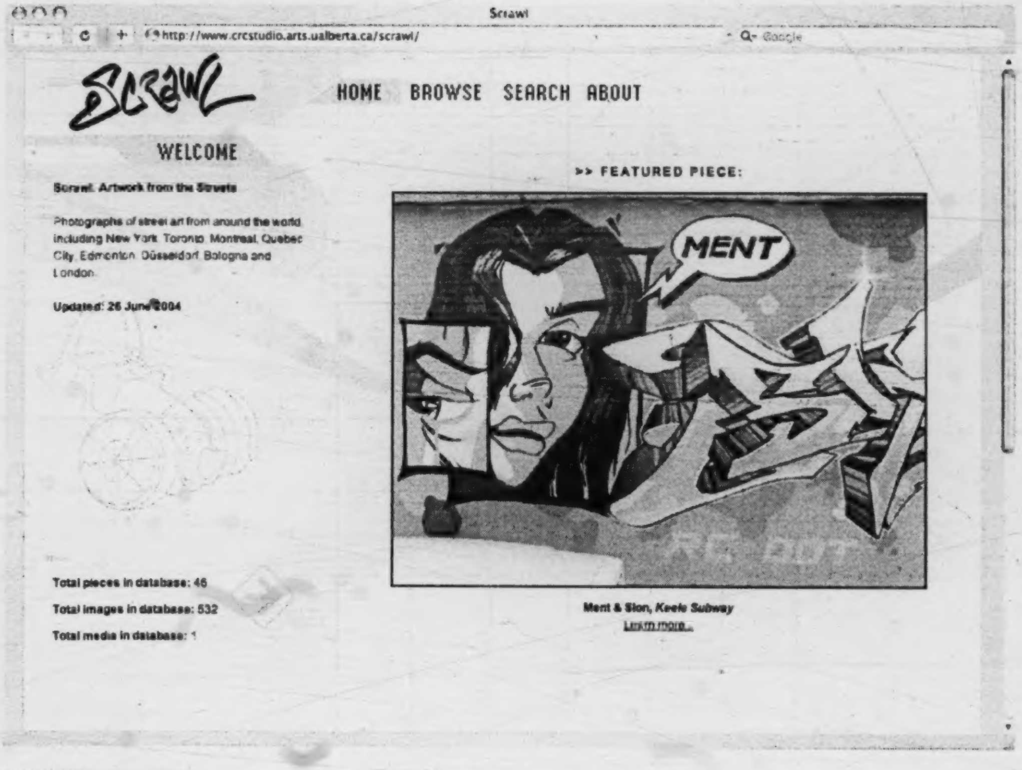
This website is powered by the [illegible] and can be easily [illegible] or [illegible] in a variety of ways.

Total texts in database: 85

Total images in database: 465

Loading "http://www.crcstudio.arts.ualberta.ca/activist/", completed 16 of 17 items

Figure A-30
Edmonton Activist Literature
January 2004


Scrawl
http://www.crcstudio.arts.ualberta.ca/scrawl/
Google
Scrawl
HOME
BROWSE
SEARCH
ABOUT
WELCOME
>> FEATURED PIECE:
MENT
Learn more...

Scrawl: Artwork from the Streets

Photographs of street art from around the world including New York, Toronto, Montreal, Quebec City, Edmonton, Düsseldorf, Bologna and London.

Updated: 26 June 2004

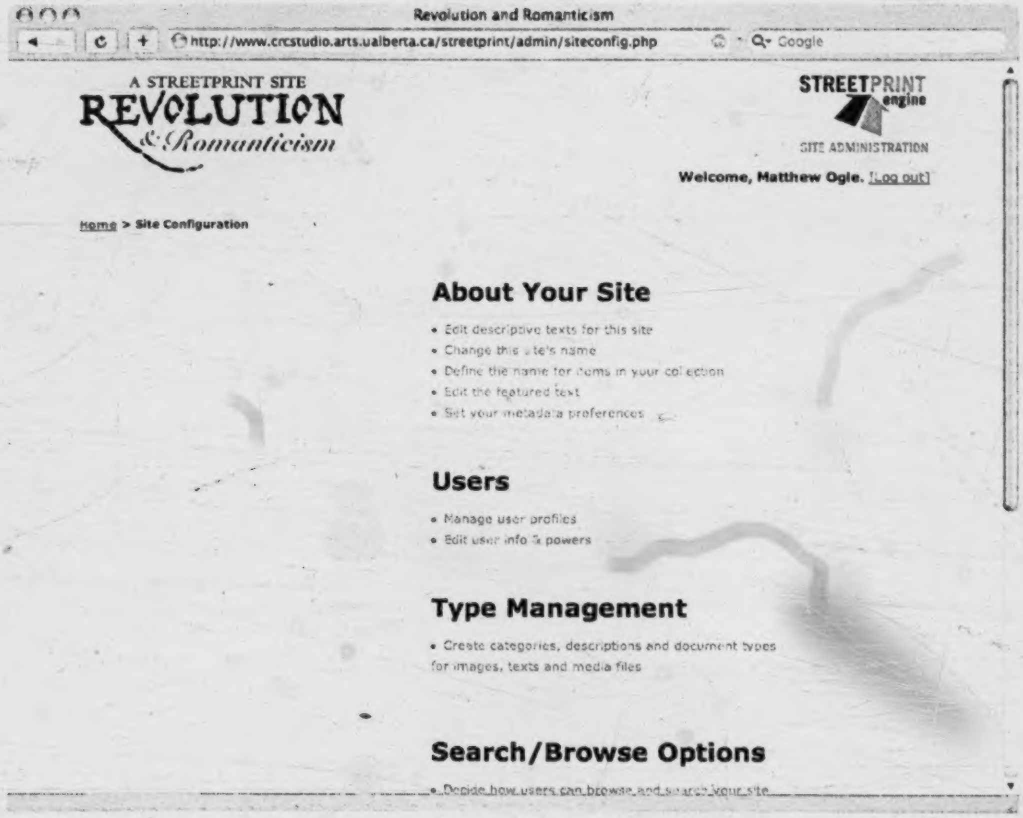
Total pieces in database: 46

Total images in database: 532

Total media in database: 1

Ment & Sion, *Keele Subway*

Figure A-31
Scrawl: Artwork from the Streets
January 2004



Revolution and Romanticism
http://www.crcstudio.arts.ualberta.ca/streetprint/admin/siteconfig.php
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
STREETPRINT
engine
SITE ADMINISTRATION

Welcome, Matthew Ogle. [Log out]

Home > **Site Configuration**

About Your Site

- Edit descriptive texts for this site
- Change this site's name
- Define the name for items in your collection
- Edit the featured text
- Set your metadata preferences

Users

- Manage user profiles
- Edit user info & powers

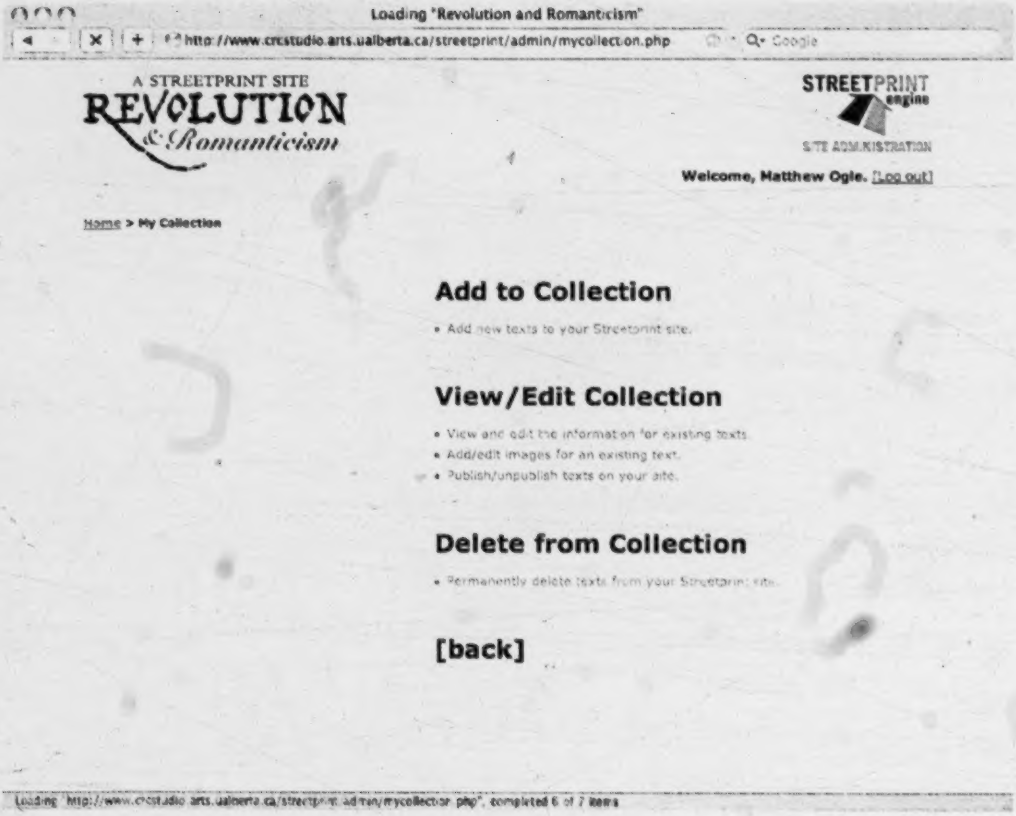
Type Management

- Create categories, descriptions and document types for images, texts and media files

Search/Browse Options

- Decide how users can browse and search your site

Figure A-32
Streetprint 2.0 Administrative Tools
Site Configuration Screen



Loading "Revolution and Romanticism"
http://www.crcstudio.arts.ualberta.ca/streetprint/admin/mycollection.php
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
STREETPRINT engine
SITE ADMINISTRATION

Welcome, Matthew Ogle. [Log out]

Home > My Collection

Add to Collection

- Add new texts to your Streetprint site.

View/Edit Collection

- View and edit the information for existing texts.
- Add/edit images for an existing text.
- Publish/unpublish texts on your site.

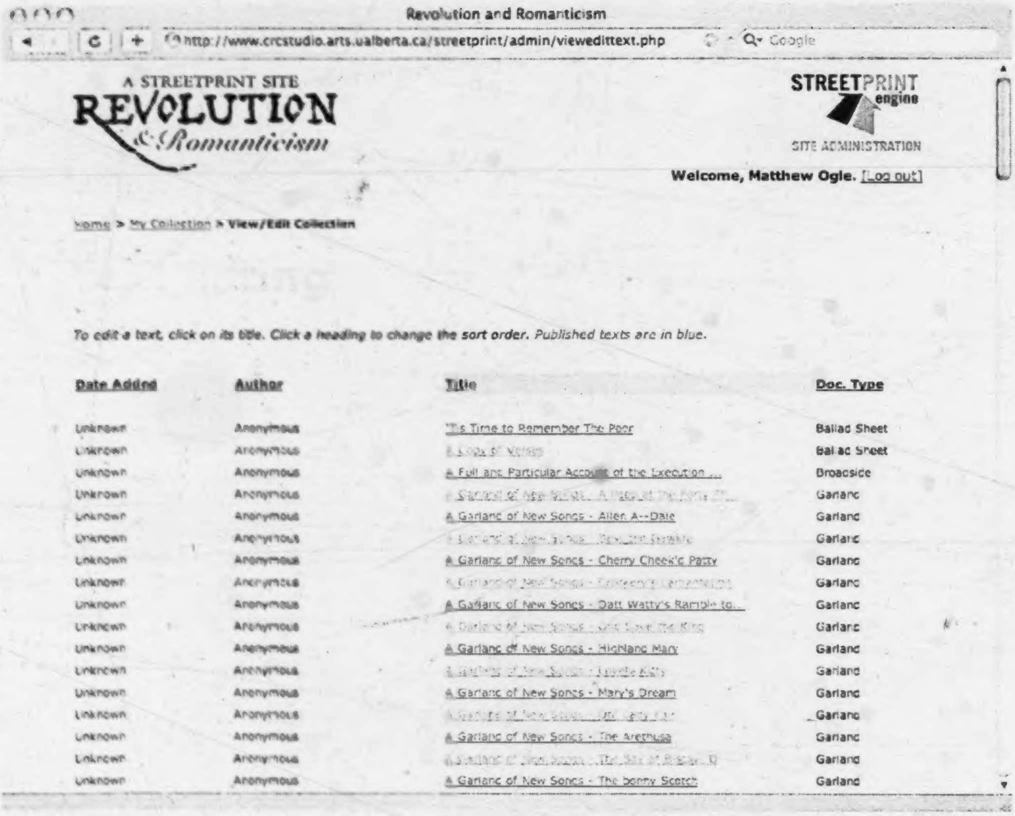
Delete from Collection

- Permanently delete texts from your Streetprint site.

[back]

Loading "http://www.crcstudio.arts.ualberta.ca/streetprint/admin/mycollection.php", completed 6 of 7 items

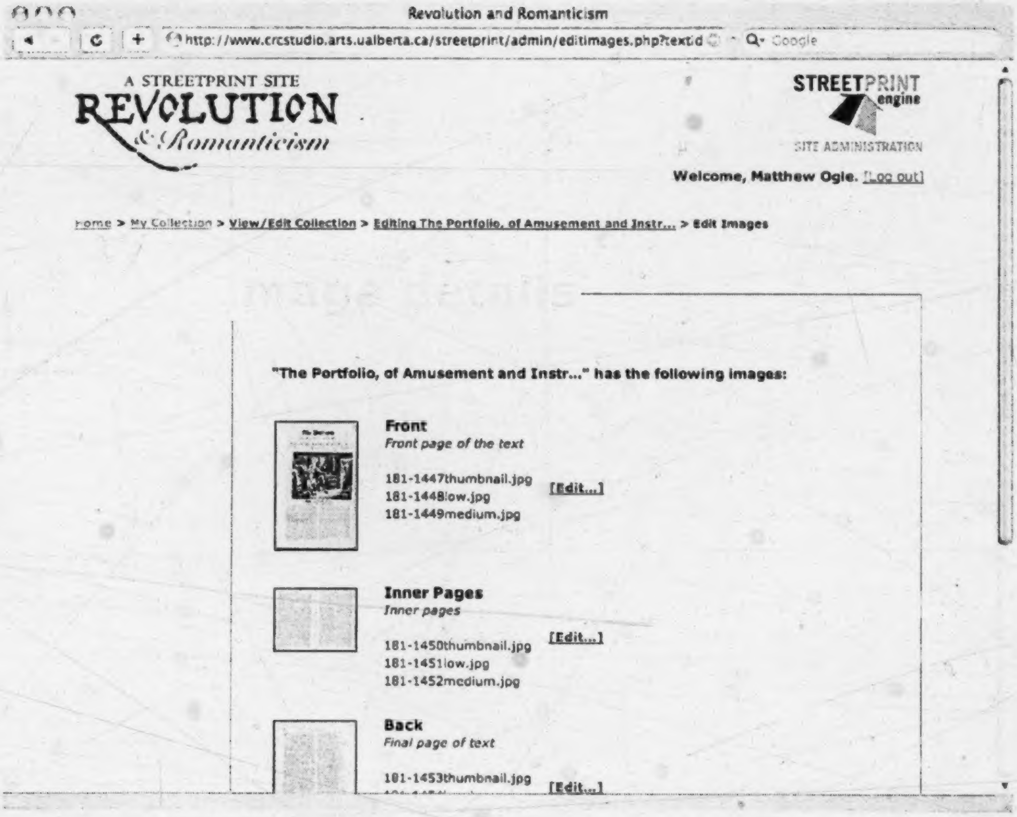
Figure A-33
Streetprint 2.0 Administrative Tools
My Collection Screen


Revolution and Romanticism
http://www.crcstudio.arts.ualberta.ca/streetprint/admin/viewedittext.php
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
STREETPRINT engine
SITE ADMINISTRATION
Welcome, Matthew Ogle. [Log out]
Home > My Collection > View/Edit Collection

To edit a text, click on its title. Click a heading to change the sort order. Published texts are in blue.

Date Added	Author	Title	Doc. Type
Unknown	Anonymous	'Tis Time to Remember The Poor	Ballad Sheet
Unknown	Anonymous	A Copy of Verses	Ballad Sheet
Unknown	Anonymous	A Full and Particular Account of the Execution ...	Broadside
Unknown	Anonymous	A Garland of New Songs - [illegible]	Garland
Unknown	Anonymous	A Garland of New Songs - Allen A--Dale	Garland
Unknown	Anonymous	A Garland of New Songs - [illegible]	Garland
Unknown	Anonymous	A Garland of New Songs - Cherry Cheek'd Patty	Garland
Unknown	Anonymous	A Garland of New Songs - [illegible]	Garland
Unknown	Anonymous	A Garland of New Songs - Datt Watty's Ramble to...	Garland
Unknown	Anonymous	A Garland of New Songs - God Save the King	Garland
Unknown	Anonymous	A Garland of New Songs - Highland Mary	Garland
Unknown	Anonymous	A Garland of New Songs - [illegible]	Garland
Unknown	Anonymous	A Garland of New Songs - Mary's Dream	Garland
Unknown	Anonymous	A Garland of New Songs - [illegible]	Garland
Unknown	Anonymous	A Garland of New Songs - The Arethusa	Garland
Unknown	Anonymous	A Garland of New Songs - The Bay of Biscay, O	Garland
Unknown	Anonymous	A Garland of New Songs - The bonny Scotch	Garland

Figure A-33
Streetprint 2.0 Administrative Tools
Text Listing



Revolution and Romanticism
http://www.crcstudio.arts.ualberta.ca/streetprint/admin/editimages.php?textid
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
STREETPRINT engine
SITE ADMINISTRATION

Welcome, Matthew Ogle. [Log out]

Home > My Collection > View/Edit Collection > Editing The Portfolio, of Amusement and Instr... > Edit Images

"The Portfolio, of Amusement and Instr..." has the following images:

Front
Front page of the text

181-1447thumbnail.jpg
181-1448low.jpg
181-1449medium.jpg
[Edit...]

Inner Pages
Inner pages

181-1450thumbnail.jpg
181-1451low.jpg
181-1452medium.jpg
[Edit...]

Back
Final page of text

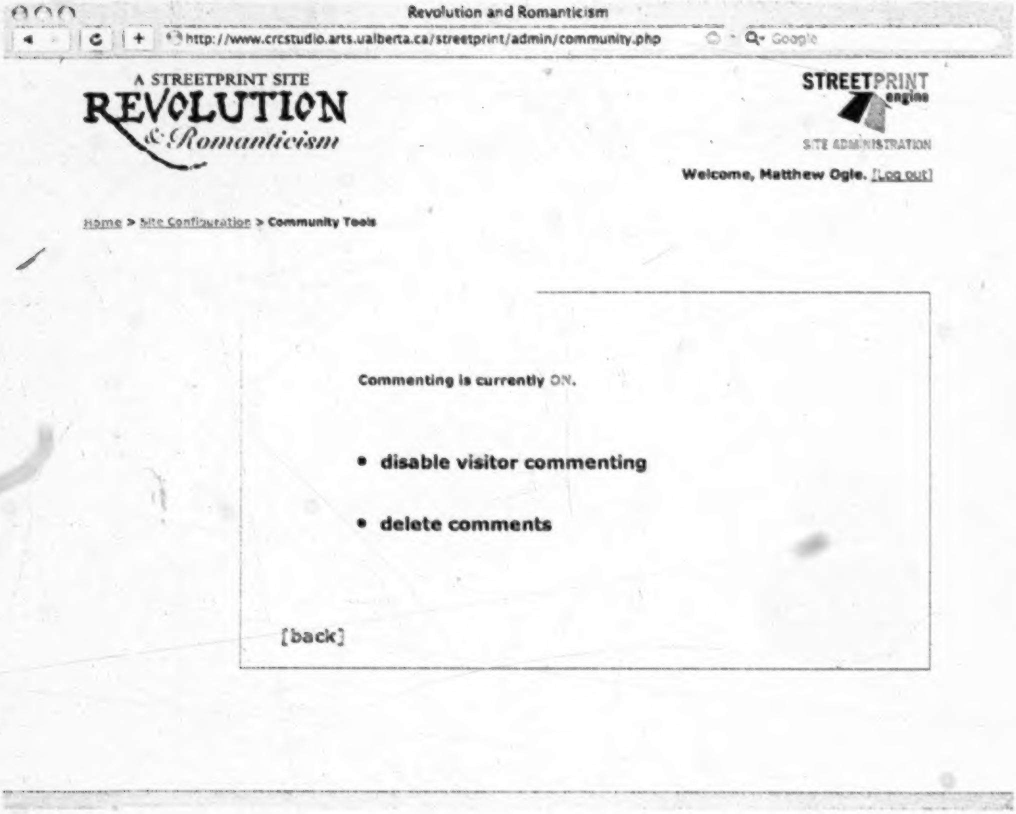
181-1453thumbnail.jpg
[Edit...]

Figure A-34
Streetprint 2.0 Administrative Tools
Image Details Screen

```
# STREETPRINT 2.0 => 2.1 UPGRADE SQL
# 16 April 2004

# SP 2.1 ADDITIONS

# Comments table

CREATE TABLE comments (

  id int(11) NOT NULL auto_increment,

  text_id int(11) NOT NULL default '0',
  commenter TEXT,
  email TEXT,
  comment TEXT,
  timestamp DATETIME,
  adminviewed INT,

  PRIMARY KEY  (id),

  FOREIGN KEY (text_id) REFERENCES texts(id)

) TYPE=MyISAM;


# The next three lines are to upgrade 2.0 installs only:

alter table spconfig add column commenting_on int;
alter table spconfig add column metadata int;
update spconfig set metadata='1';
```

Figure A-35

Streetprint 2.1 SQL Upgrade Commands

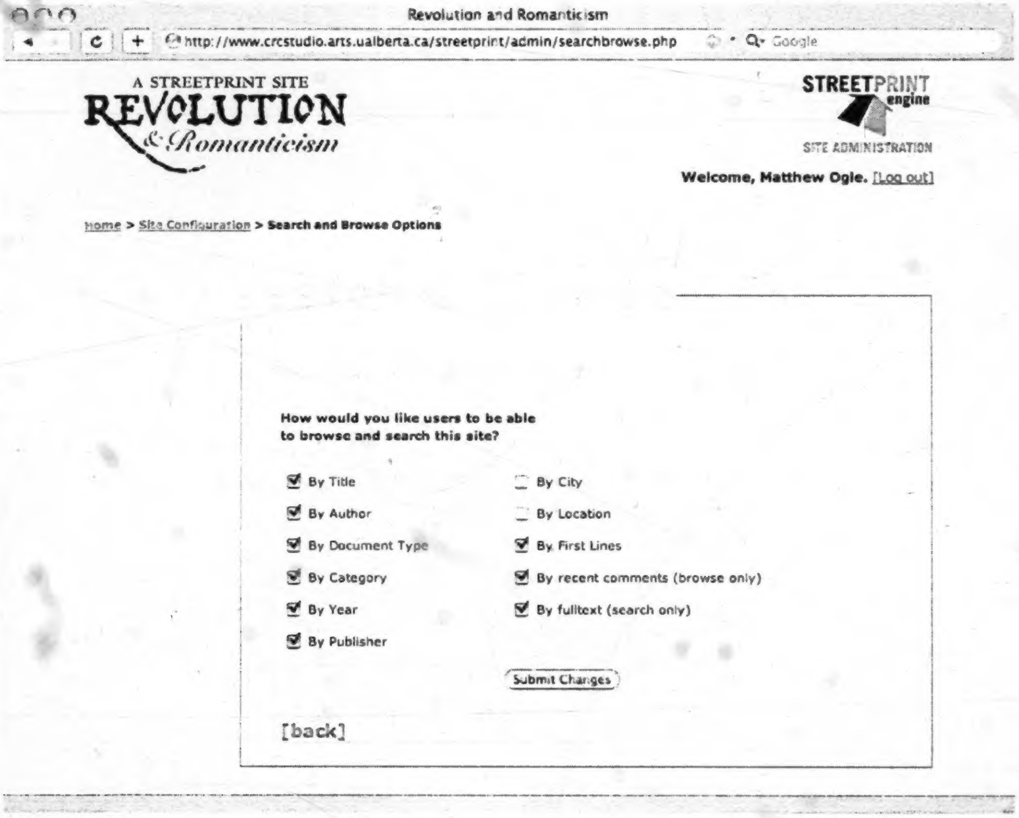

Revolution and Romanticism
http://www.crcstudio.arts.ualberta.ca/streetprint/admin/community.php
Google
A STREETPRINT SITE
REVOLUTION & Romanticism
STREETPRINT engine
SITE ADMINISTRATION
Welcome, Matthew Ogle. [Log out]
Home > Site Configuration > Community Tools

Commenting is currently ON.

- disable visitor commenting
- delete comments

[back]

Figure A-36
Streetprint 2.1 Administrative Tools
Community Screen


Revolution and Romanticism
http://www.crcstudio.arts.ualberta.ca/streetprint/admin/searchbrowse.php
Google
A STREETPRINT SITE
REVOLUTION
& Romanticism
STREETPRINT engine
SITE ADMINISTRATION

Welcome, Matthew Ogle. [Log out]

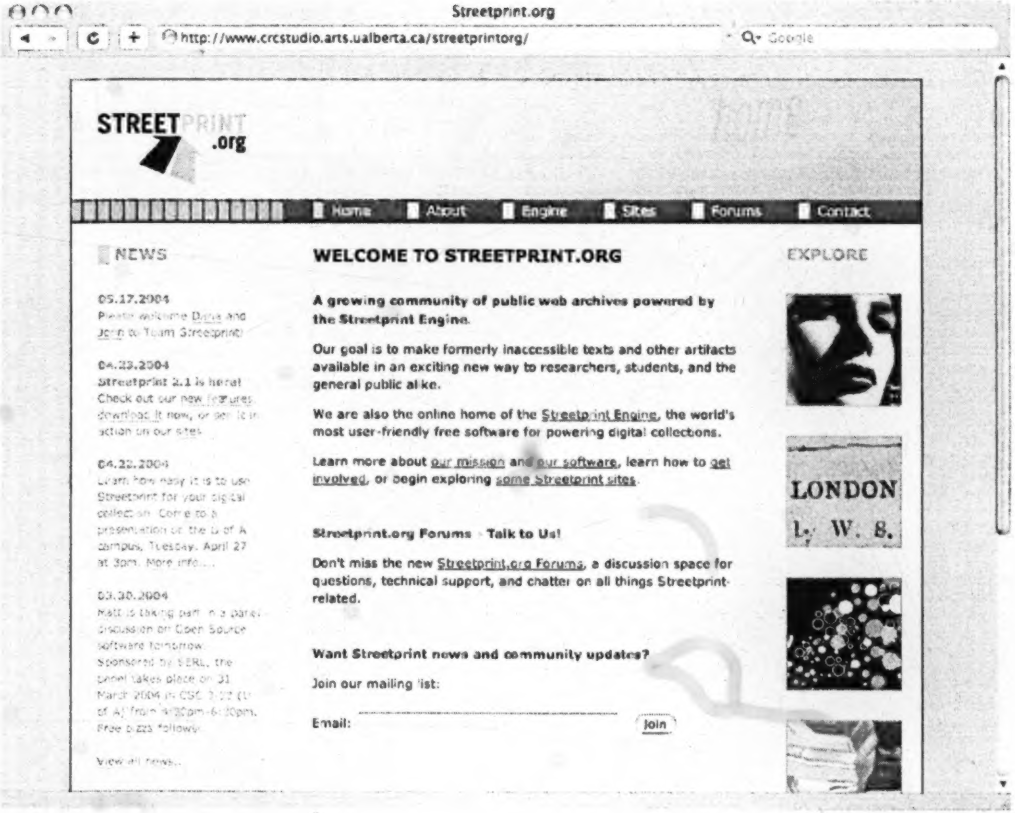
Home > Site Configuration > Search and Browse Options

How would you like users to be able to browse and search this site?

☑ By Title
☑ By Author
☑ By Document Type
☑ By Category
☑ By Year
☑ By Publisher
☐ By City
☐ By Location
☑ By First Lines
☑ By recent comments (browse only)
☑ By fulltext (search only)

Submit Changes
[back]

Figure A-37
Streetprint 2.1 Administrative Tools
Search and Browse Options


Streetprint.org
http://www.crcstudio.arts.ualberta.ca/streetprintorg/
Google
STREETPRINT .org
Home
About
Engine
Sites
Forums
Contact
NEWS
EXPLORE
LONDON
L. W. S.

05.17.2004
Please welcome Dana and John to Team Streetprint!

04.23.2004
Streetprint 2.1 is here! Check out our new features, download it now, or see it in action on our sites.

04.22.2004
Learn how easy it is to use Streetprint for your digital collection. Come to a presentation on the U of A campus, Tuesday, April 27 at 3pm. More info....

03.30.2004
Matt is taking part in a panel discussion on Open Source software tomorrow. Sponsored by SERL, the panel takes place on 31 March 2004 in CSC [illegible] (U of A) from 4:30pm-6:30pm. Free pizza follows.

View all news...

WELCOME TO STREETPRINT.ORG

A growing community of public web archives powered by the Streetprint Engine.

Our goal is to make formerly inaccessible texts and other artifacts available in an exciting new way to researchers, students, and the general public alike.

We are also the online home of the Streetprint Engine, the world's most user-friendly free software for powering digital collections.

Learn more about our mission and our software, learn how to get involved, or begin exploring some Streetprint sites.

Streetprint.org Forums - Talk to Us!

Don't miss the new Streetprint.org Forums, a discussion space for questions, technical support, and chatter on all things Streetprint-related.

Want Streetprint news and community updates?

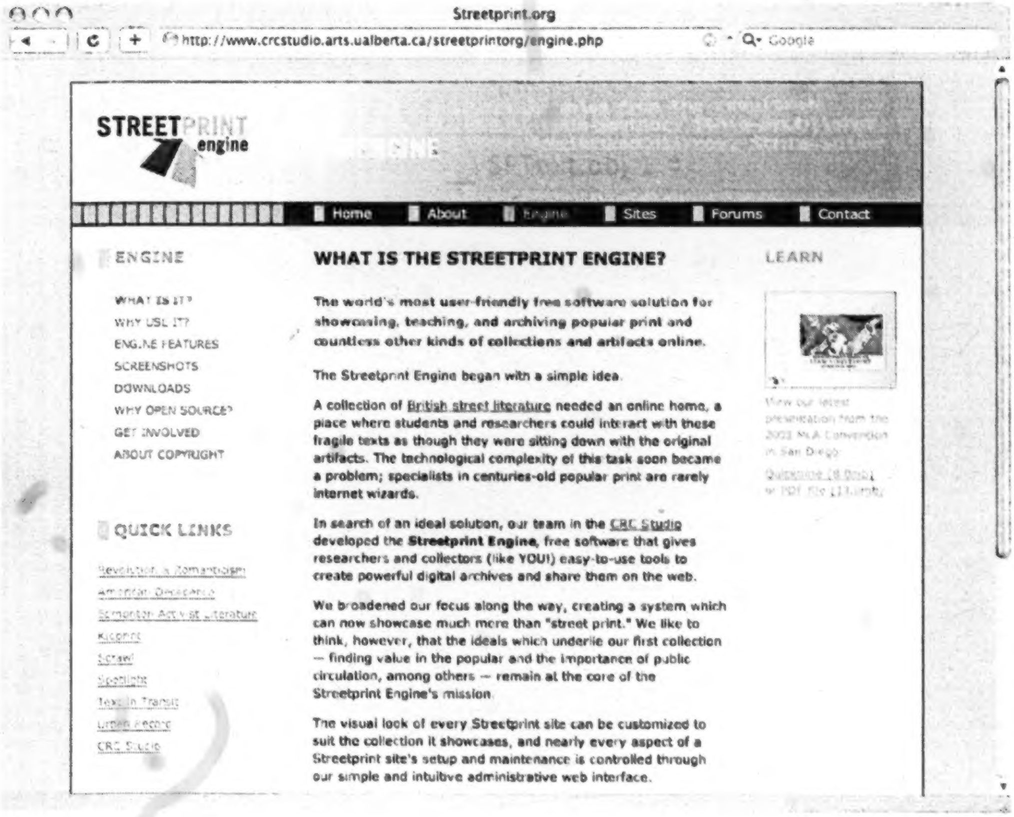
Join our mailing list:

Email: Join

Figure A-38

Streetprint.org, front page

July 2004


Streetprint.org
http://www.crcstudio.arts.ualberta.ca/streetprintorg/engine.php
Google
STREETPRINT engine
Home
About
Engine
Sites
Forums
Contact
ENGINE
WHAT IS IT?
WHY USE IT?
ENGINE FEATURES
SCREENSHOTS
DOWNLOADS
WHY OPEN SOURCE?
GET INVOLVED
ABOUT COPYRIGHT
QUICK LINKS
Revolution & Romanticism
American Dissidence
Edmonton Activist Literature
Kidprint
Scrawl
Spotlight
Text in Transit
Urban Record
CRC Studio
LEARN

WHAT IS THE STREETPRINT ENGINE?

The world's most user-friendly free software solution for showcasing, teaching, and archiving popular print and countless other kinds of collections and artifacts online.

The Streetprint Engine began with a simple idea.

A collection of British street literature needed an online home, a place where students and researchers could interact with these fragile texts as though they were sitting down with the original artifacts. The technological complexity of this task soon became a problem; specialists in centuries-old popular print are rarely internet wizards.

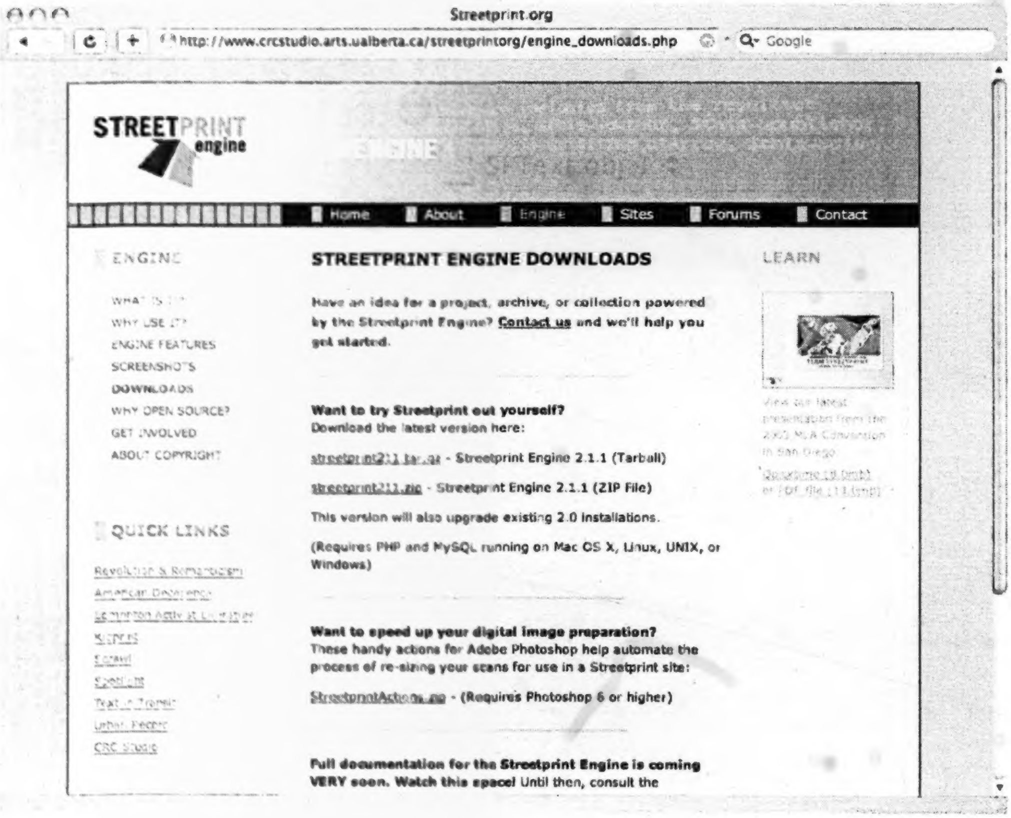
In search of an ideal solution, our team in the CRC Studio developed the **Streetprint Engine**, free software that gives researchers and collectors (like YOU!) easy-to-use tools to create powerful digital archives and share them on the web.

We broadened our focus along the way, creating a system which can now showcase much more than "street print." We like to think, however, that the ideals which underlie our first collection — finding value in the popular and the importance of public circulation, among others — remain at the core of the Streetprint Engine's mission.

The visual look of every Streetprint site can be customized to suit the collection it showcases, and nearly every aspect of a Streetprint site's setup and maintenance is controlled through our simple and intuitive administrative web interface.

View our latest presentation from the 2003 MLA Convention in San Diego: Quicktime [illegible] or PDF file [illegible]

Figure A-39
Streetprint.org, Engine page
July 2004


Streetprint.org
http://www.crcstudio.arts.ualberta.ca/streetprintorg/engine_downloads.php
Google
STREETPRINT engine
Home
About
Engine
Sites
Forums
Contact
ENGINE
WHAT IS IT?
WHY USE IT?
ENGINE FEATURES
SCREENSHOTS
DOWNLOADS
WHY OPEN SOURCE?
GET INVOLVED
ABOUT COPYRIGHT
QUICK LINKS
Revolution & Romanticism
American Decadence
LEARN

STREETPRINT ENGINE DOWNLOADS

Have an idea for a project, archive, or collection powered by the Streetprint Engine? Contact us and we'll help you get started.

Want to try Streetprint out yourself?
Download the latest version here:

streetprint211.tar.gz - Streetprint Engine 2.1.1 (Tarball)

streetprint211.zip - Streetprint Engine 2.1.1 (ZIP File)

This version will also upgrade existing 2.0 installations.

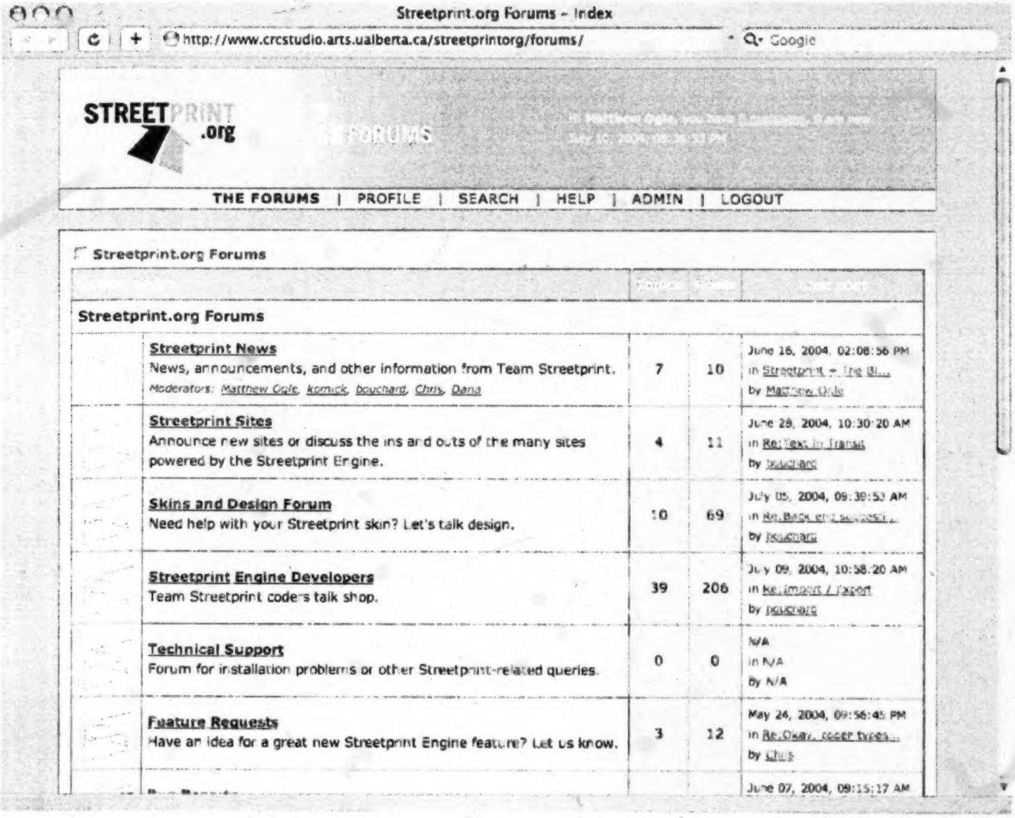
(Requires PHP and MySQL running on Mac OS X, Linux, UNIX, or Windows)

Want to speed up your digital image preparation?
These handy actions for Adobe Photoshop help automate the process of re-sizing your scans for use in a Streetprint site:

StreetprintActions.zip - (Requires Photoshop 6 or higher)

Full documentation for the Streetprint Engine is coming VERY soon. Watch this space! Until then, consult the

Figure A-40
Streetprint.org, Engine downloads page
July 2004



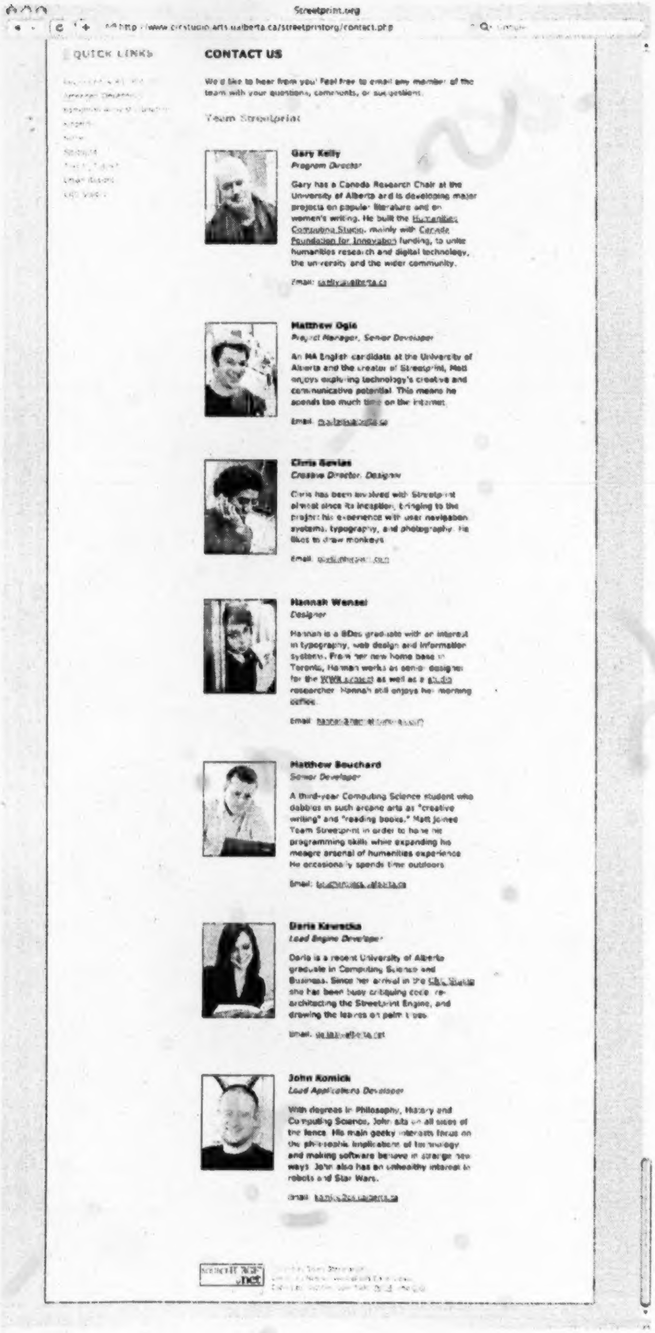
Streetprint.org Forums – Index

http://www.crcstudio.arts.ualberta.ca/streetprintorg/forums/

Streetprint.org Forums

Streetprint.org Forums			
Streetprint News News, announcements, and other information from Team Streetprint. *Moderators: Matthew Ogle, kornick, bouchard, Chris, Dana*	7	10	June 16, 2004, 02:08:56 PM in Streetprint + The Bl... by Matthew Ogle
Streetprint Sites Announce new sites or discuss the ins and outs of the many sites powered by the Streetprint Engine.	4	11	June 28, 2004, 10:30:20 AM in Re: Text in Transit by bouchard
Skins and Design Forum Need help with your Streetprint skin? Let's talk design.	10	69	July 05, 2004, 09:39:53 AM in Re: Back end success... by bouchard
Streetprint Engine Developers Team Streetprint coders talk shop.	39	206	July 09, 2004, 10:58:20 AM in Re: import / export by bouchard
Technical Support Forum for installation problems or other Streetprint-related queries.	0	0	N/A in N/A by N/A
Feature Requests Have an idea for a great new Streetprint Engine feature? Let us know.	3	12	May 24, 2004, 09:56:45 PM in Re: Okay, coder types... by Chris
			June 07, 2004, 09:15:17 AM

Figure A-41
Streetprint.org, Forums page
July 2004


Streetprint.org
http://www.crcstudio.arts.ualberta.ca/streetprintorg/contact.php
QUICK LINKS

CONTACT US

We'd like to hear from you! Feel free to email any member of the team with your questions, comments, or suggestions.

Team Streetprint

Gary Kelly
Program Director

Gary has a Canada Research Chair at the University of Alberta and is developing major projects on popular literature and on women's writing. He built the Humanities Computing Studio, mainly with Canada Foundation for Innovation funding, to unite humanities research and digital technology, the university and the wider community.

Email: gkelly@ualberta.ca

Matthew Ogle
Project Manager, Senior Developer

An MA English candidate at the University of Alberta and the creator of Streetprint, Matt enjoys exploring technology's creative and communicative potential. This means he spends too much time on the internet.

Email: [illegible]

Chris Bevies
Creative Director, Designer

Chris has been involved with Streetprint almost since its inception, bringing to the project his experience with user navigation systems, typography, and photography. He likes to draw monkeys.

Email: [illegible]

Hannah Wensel
Designer

Hannah is a BDes graduate with an interest in typography, web design and information systems. From her new home base in Toronto, Hannah works as senior designer for the WWR project as well as a studio researcher. Hannah still enjoys her morning coffee.

Email: [illegible]

Matthew Bouchard
Senior Developer

A third-year Computing Science student who dabbles in such arcane arts as "creative writing" and "reading books." Matt joined Team Streetprint in order to hone his programming skills while expanding his meagre arsenal of humanities experience. He occasionally spends time outdoors.

Email: [illegible]

Daria Kowalczyk
Lead Engine Developer

Daria is a recent University of Alberta graduate in Computing Science and Business. Since her arrival in the CRC Studio she has been busy critiquing code, re-architecting the Streetprint Engine, and drawing the leaves on palm trees.

Email: [illegible]

John Komick
Lead Applications Developer

With degrees in Philosophy, History and Computing Science, John sits on all sides of the fence. His main geeky interests focus on the philosophic implications of technology and making software behave in strange new ways. John also has an unhealthy interest in robots and Star Wars.

Email: [illegible]

Figure A-42

Team Streetprint, July 2004

```
# STREETPRINT UPGRADE SQL
# July2004

# SP 3.x ADDITIONS

alter table texts add column conf_notes text;
alter table texts add column ref_num int;
alter table texts add column date date;

alter table texts add column num_field1 int;
alter table texts add column num_field2 int;
alter table texts add column num_field3 int;
alter table texts add column num_field4 int;
alter table texts add column num_field5 int;

alter table texts add column date_field1 date;
alter table texts add column date_field2 date;
alter table texts add column date_field3 date;

alter table texts add column text_field1 text;
alter table texts add column text_field2 text;
alter table texts add column text_field3 text;
alter table texts add column text_field4 text;
alter table texts add column text_field5 text;
alter table texts add column text_field6 text;
alter table texts add column text_field7 text;
alter table texts add column text_field8 text;
alter table texts add column text_field9 text;
alter table texts add column text_field10 text;
```

Figure A-43

Tentative Upgrade SQL for Streetprint 3.x

Appendix B

This appendix contains a complete copy of the user manual for version 2.1 of the Streetprint Engine.

The manual's pages have been printed in black and white and reduced to 85% of their normal size in order to accommodate this document's margins and pagination.

A full-sized colour PDF version of this manual is available online at *www.streetprint.org.*

STREET 1

VERSION 2.1 USER MANUAL

Streetprint Engine 2.1 User Manual
by Matthew Ogle

(cc) 2004 Streetprint.org

Some rights reserved. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/2.0/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

The events and characters portrayed in this user manual's case study are fictitious. Any resemblance to persons, living or dead, is purely concidental.

This manual was designed and set into type by Hannah Wensel. Illustrations by Chris Govias. Printed and published in Edmonton, Alberta at the CRC Humanities Computing Studio (University of Alberta)

The text face is Fairfield Light, designed by Rudolph Růžička for Linotype in 1939, digitized by Alex Kazcun in 1991.

The captions are set in Univers Condensed, designed by Adrian Frutiger in 1957 for Deberny & Piegnot, digitized by Linotype in 1997.

VERSION 2.1 USER MANUAL

STREETPRINT engine

Contents

5 **I. Introduction**
  6 Preface
  7 Old Man Grinnel and the Milk River Archive

9 **II. Installation**
  10 PHP and MySQL
  10 Download
  11 Install

13 **III. Site Configuration**
  14 The Administrative Interface
  15 Getting Started
  17 Users
  18 About Your Site
  21 Type Management
  27 Search/Browse Options
  28 Community Tools
  30 Course/Syllabus Tools

31 **IV. Managing Your Collection**
  32 Overview
  33 Garbage In, Garbage Out
  35 Image Capture
  36 Image Processing
  42 Adding a Text Record
  48 Adding Images
  52 Publishing Your Item
  54 Editing Your Collection
  55 Deleting From Your Collection
  56 That's All, Folks

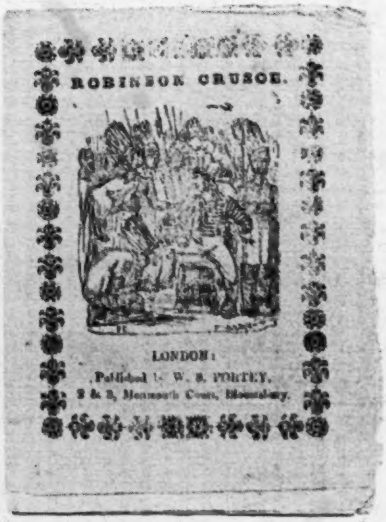
57 **V. Skins**
  58 Comfortable In Your Skin
  59 The Graphics Directory
  60 Roll Your Own
  61 Finishing Up
  63 Welcome Aboard

CHAPTER I

INTRODUCTION

Preface

The Streetprint Engine began with a simple idea.



ROBINSON CRUSOE.

LONDON:
Published by W. S. FORTEY,

Fig 1. **Robinson Crusoe, from Streetprint: "Revolution and Romanticism"**

A collection of British street literature needed an online home, a place where students and researchers could interact with these fragile texts as though they were sitting down with the original artifacts. The technological complexity of this task soon became a problem; specialists in centuries-old popular print are rarely internet wizards. In search of an ideal solution, our team in the CRC Studio *(www.crcstudio.arts.ualberta.ca)* developed the Streetprint Engine, free software that gives researchers and collectors (like you) easy-to-use tools to create powerful digital archives and share them on the web. Team Streetprint broadened its focus along the way, creating a system which can now showcase much more than "street print." We like to think, however, that the ideals which underlie our first collection—finding value in the popular and the importance of public circulation, among others—remain at the core of the Streetprint Engine's mission.

The next few pages of this manual will introduce you to "Old Man Grinnel" and lead you through the process of creating a Streetprint site which showcases his eccentric (and imaginary) collection. Along the way, you may be surprised when we point out our software's rough edges in addition to the tasks it handles with ease.

Our intention with this manual's case study is to push the boundaries of our current Streetprint Engine version, illuminating its flexibility but also its current restrictions. Why do this? Since the Streetprint Engine is "open source" software, any programmer in the world can view our code and make improvements which will then benefit all Streetprint users. Even if you're not a programmer, you can join the team by reporting bugs or letting us know what improvements you would like to see in the next version of the Streetprint Engine. By highlighting some of our current shortcomings, we hope that we can spur development and turn any existing weaknesses into future areas of strength

We hope you enjoy this manual. See you online!

– *Team Streetprint (www.streetprint.org)*

Old Man Grinnel and the Milk River Archive

> “*Scattered among these masquerades were people of a different type...they loitered on the corners or stood with their backs to the store windows and stared at everyone who passed...Tod knew very little about them except that they had come to California to die.*”
>
> —Nathanael West, *The Day of the Locust*

Kasper von Grinnel left Milk River much the same way he came; quietly, mysteriously, unceremoniously. After nearly forty years on an old homestead near this southern Alberta town, the strange recluse from California was a fixture in Milk River lore. Old Man Grinnel had faked his own death back in the 1950s, the locals swore, fled north from Hollywood during the communist scare. Neighbours claimed he hadn't left his house in years; children dared each other to throw rocks through his farmhouse windows on Halloween night. And then one day he was gone.

It took a while for Milk River to notice. Grinnel's farm gradually went to seed, the grocer wondered why the phone failed to ring with the usual weekly order. When the local RCMP detachment entered the house and found no body, no trace of a struggle, new rumors swirled about town: Grinnel was on the run again; he'd gone back to California; he'd been stealthily abducted. But as time passed, it became clear that the old eccentric was gone for good, and his farmhouse was put up for auction.

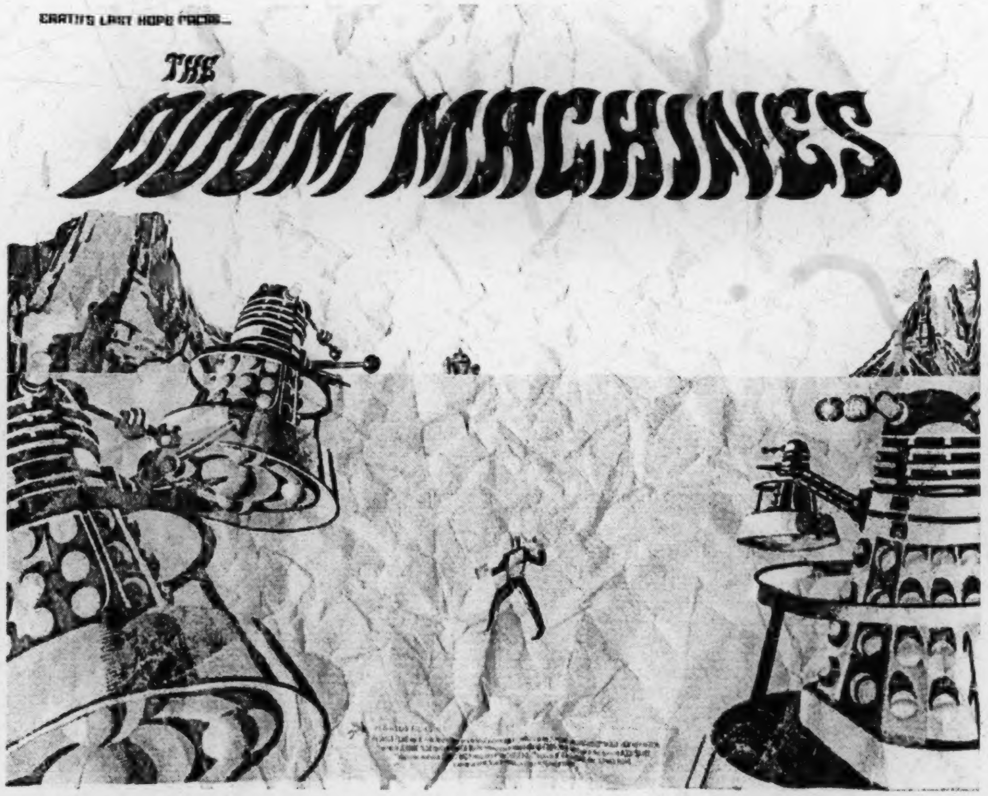
It was around that time that an RCMP officer came knocking at your door. Barely into your fourth week as Milk River's first official civic archivist, you were surprised to hear the officer had something to show you. No corpse had been found in Grinnel's farmhouse, but something else had: the old man's sprawling collection of Hollywood memorabilia. The cops had filled an entire stockroom with the stuff: film reels and movie posters, signed photographs, letters from directors and starlets, even a bizarre assortment of old props from horror and science fiction movies. The RCMP didn't feel like storing it for much longer, the officer explained. “Do you want this stuff?”

And so your first big project has begun. The material in Grinnel's collection is compelling, providing a powerful textual and visual history of Hollywood's seedier side, and even hinting at what brought Grinnel into your own community so many years ago. But how to catalogue this collection and share your research? A digital archive would be ideal, since it would both preserve the material and allow you to share it with the greater public, but you're not a computer expert and the town's archival budget is meagre at best. You occupy an office along with the mayor's secretary, and your equipment consists of a single computer, a digital camera, a scanner with some image-editing software, and an internet connection. What can be done?

Enter the Streetprint Engine. This manual will guide you, step-by-step, on how to use our free software to create a website which showcases and preserves Kasper von Grinnel's eccentric legacy. Let's get started...

Look out! It's the Doom Machines!

Fig 2. Poster for The 'Doom Machines' from Kasper von Grinnel's collection



CHAPTER II

INSTALLATION

PHP and MySQL

Geek Alert!
This chapter of the Streetprint Engine manual is by far the geekiest of the lot. If you need any help with this section, we recommend you consult your nearest internet wizard or technical support staff member.

Before getting started with the Streetprint Engine, you need access to a web server which can host your Streetprint site and give it a home on the Internet. Streetprint requires a web server with the following software installed:

- PHP 4.1 or higher*
- MySQL 3.23 or higher**

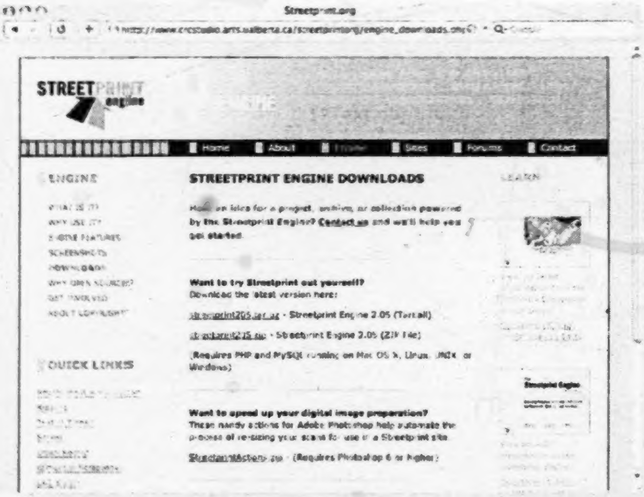
This configuration is a common one; nearly all web hosting companies should be able to provide this setup for only a small monthly fee. Alternatively, you can run the Streetprint Engine from your own computer running Mac OS X, Linux, or Windows provided that you are able to install and configure web server software with PHP and MySQL.

* PHP support can be found at *www.php.net*
** MySQL support at *www.mysql.com*

Download

Once the server is ready—lucky us, we were able to get *www.oldmangrinnel.com* for only $8/month!—it's time to download the Streetprint Engine. It can be found online at:

www.streetprint.org/engine_downloads.php



STREETPRINT ENGINE DOWNLOADS

Want to try Streetprint out yourself?
Download the latest version here:

Want to speed up your digital image preparation?

Fig 3. The download page at Streetprint.org.

Linux/UNIX users should grab the .tar.gz file, and Mac/Windows users should use the .zip file.

Install

When your download is complete, easy installation instructions can be found in the included ReadMe.html file. Follow the on-screen directions and your Streetprint site will be up and running in a matter of minutes!

(There are unfortunately so many different server configurations that it is impossible to illustrate them all here. If you have any problems with this part of the installation, we recommend contacting your web hosting company, your server administrator, or a local internet geek—they have the information you will need to complete the installation process.)

Fig 4. Uploading files with FTP.

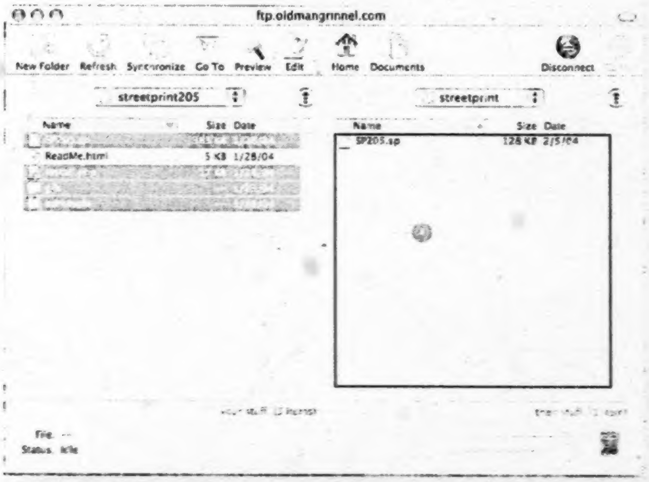
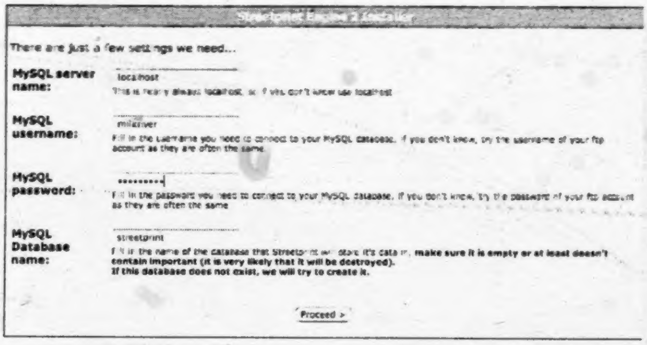

ftp.oldmangrinnel.com
New Folder
Refresh
Synchronize
Go To
Preview
Edit
Home
Documents
Disconnect
streetprint205
streetprint
Name
Size
Date
ReadMe.html
5 KB
1/28/04
SP205.sp
128 KB
2/5/04
File: --
Status: Idle

Fig 5. Setting up the database.


Streetprint Engine 2 Installer

There are just a few settings we need...

MySQL server name: localhost

This is nearly always localhost, so if you don't know use localhost

MySQL username: mikriver

Fill in the username you need to connect to your MySQL database. If you don't know, try the username of your ftp account as they are often the same

MySQL password: •••••••••

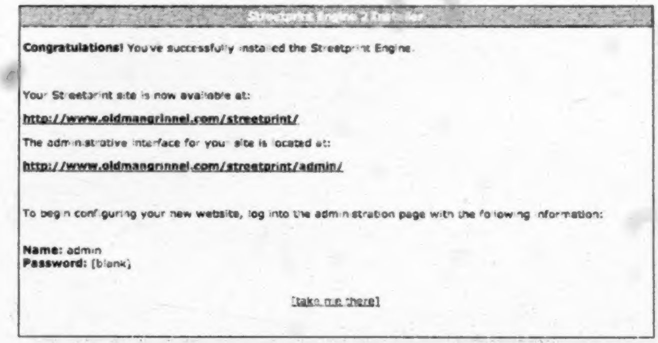
Fill in the password you need to connect to your MySQL database. If you don't know, try the password of your ftp account as they are often the same

MySQL Database name: streetprint

Fill in the name of the database that Streetprint will store it's data in, **make sure it is empty or at least doesn't contain important (it is very likely that it will be destroyed).**
If this database does not exist, we will try to create it.

Proceed >

Fig 6. All done!



Streetprint Engine 2 Installer

Congratulations! You've successfully installed the Streetprint Engine.

Your Streetprint site is now available at:

http://www.oldmangrinnel.com/streetprint/

The administrative interface for your site is located at:

http://www.oldmangrinnel.com/streetprint/admin/

To begin configuring your new website, log into the administration page with the following information:

Name: admin
Password: [blank]

[take me there]

Success! Our new Streetprint site is now online at: **www.oldmangrinnel.com/streetprint/**

For Power Users:

Streetprint's Directory structure explained

Wondering what all those files and folders in your Streetprint directory do? Here's a quick guide to some of the important ones:

admin/ Streetprint's administrative tools live in this directory.

console.inc Streetprint's navigation console, appears at the top of every page.

courses/ and coursefiles/ Course syllabi and their associated files are placed in these directories.

gfx/ This folder contains all the graphics which make up your Streetprint site's "skin." If you want to customize the design of your Streetprint site, this is the place to start.

imagelibrary/ Once images have been added to your Streetprint database, they are renamed and placed here.

medialibrary/ See above, but this time with media files.

sp_config.php This is your Streetprint site's primary configuration file. You can edit it to change database information, the names of your graphics files, your site's character set, and more.

sp_style.css Your Streetprint site's style sheet. Although future versions of Streetprint will rely more heavily on CSS-based designs, you can still make a lot of tweaks to the look of your site here.

uploadfolder/ This is where you will place images that are ready to be added to your Streetprint database.

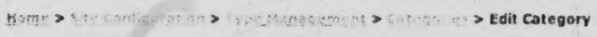
CHAPTER III

SITE CONFIGURATION

The Administrative Interface

Streetprint's administrative interface provides the tools you need to configure and maintain your Streetprint website in a friendly, easy-to-use manner. Since this manual will focus primarily on what to do with your Streetprint site (and not how to do it), here are a few quick tips for getting around:

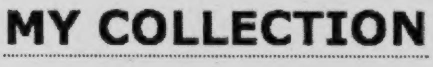
Breadcrumb Trail: Located at the top of every page, the "breadcrumb trail" works just like it does in *Hansel and Gretel;* it shows your current location and the path you took to get there. Every underlined word is a link, which can be clicked to return to a previously-visited screen.



Home > Site Configuration > Type Management > Categories > **Edit Category**

Fig 7. An example of the bread-crumb trail

Big Bold Links: It's hard to miss primary links in the Streetprint administrative interface: they're big, they're bold, and they display dotted underlines when you move your mouse over them. Clicking them will take you to a new screen.

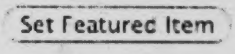


MY COLLECTION

Fig 8. Want to edit your collection? Click this

Colour Coding: Many screens in the administrative interface use colour coding: *green* for an action which will add something, *yellow* for editing, and *red* for deletion.

Click those Buttons: Changes you make on screens in Streetprint's administrative interface will not be saved until you click a button. The name of the button will remind you of the action it will perform, and you will always see a status message (in the appropriate colour) confirming your action.



Set Featured Item

Fig 9. A friendly neighbourhood button

Have Fun: You don't need to be afraid of Streetprint's administrative interface—we made it pretty difficult to break stuff. Try new things, play around, and let us know how we can continue to improve the interface in future versions of the Streetprint Engine.

Getting Started

Now that our Streetprint site is up and running, it's time to configure it specifically for Old Man Grinnel's collection.

The administrative tools for any Streetprint site can be found by adding "admin" to the website address. In our case, we would log in by going to: *www.oldmangrinnel/streetprint/admin/*

The first time you log in, use *admin* as the user name and leave the password blank. The Streetprint Engine will then prompt you to change your password and enter some personal information for your administrative account.

Once logged in, you will see the two main areas of Streetprint's administrative interface: *Site Configuration* and *My Collection*. The Site Configuration section of the Streetprint Engine is composed of six sections: *About Your Site, Users, Type Management, Search/Browse Options, Community Tools, and Course/Syllabus Tools*. This chapter of the manual will guide you through each of these sections.

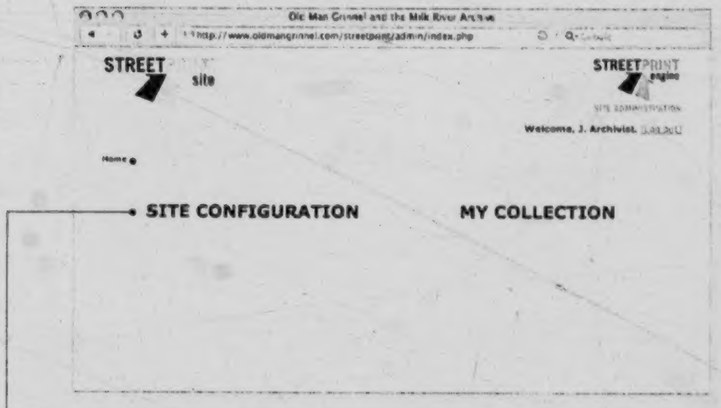
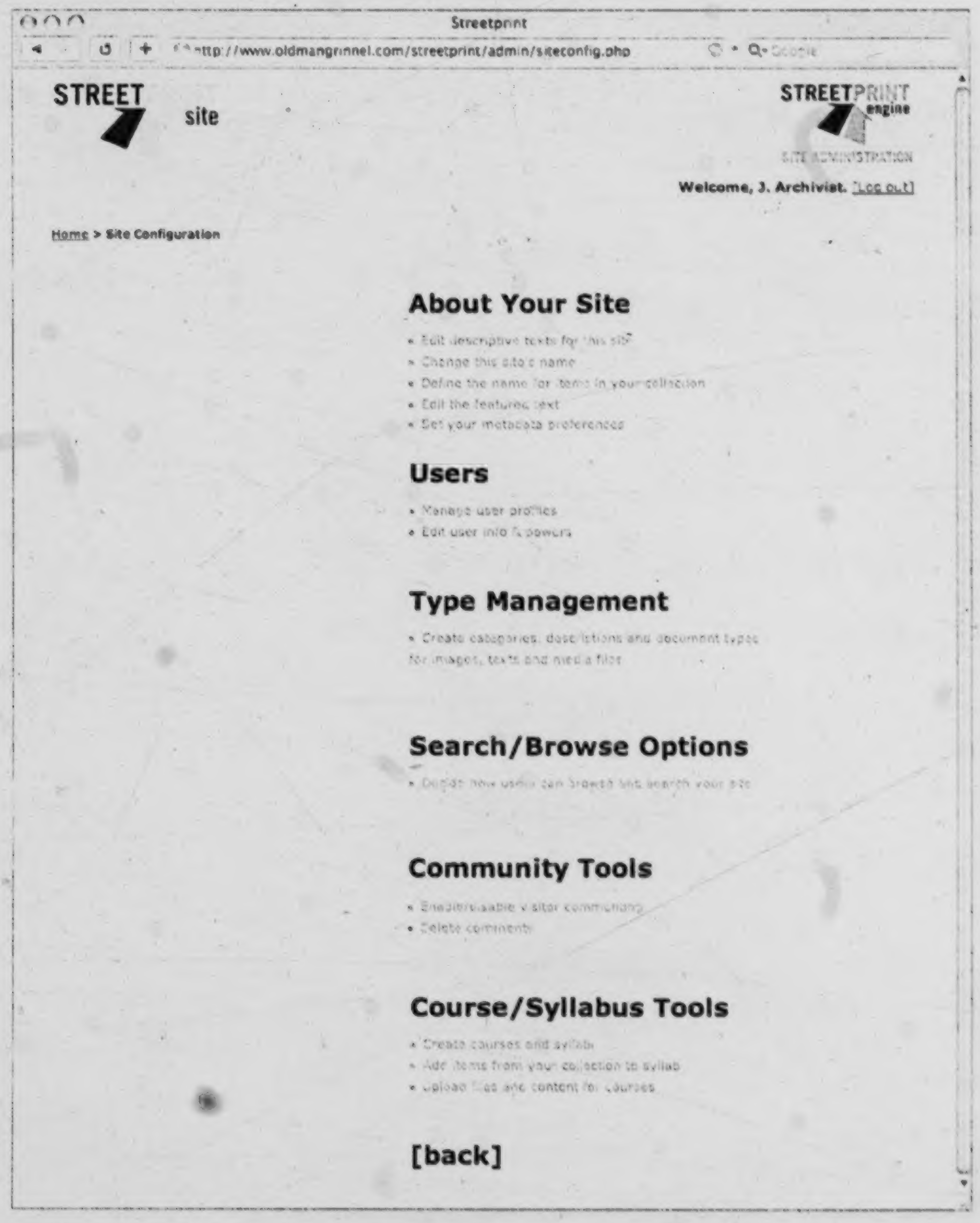

Old Man Grinnel and the Milk River Archive
http://www.oldmangrinnel.com/streetprint/admin/index.php
STREET
site
STREETPRINT
engine
SITE ADMINISTRATION
Welcome, J. Archivist.
Home
SITE CONFIGURATION
MY COLLECTION

Fig 10. The link to the Site Configuration section

Fig 11. Inside the Site Configuration section.



Streetprint
http://www.oldmangrinnel.com/streetprint/admin/siteconfig.php
STREET site
STREETPRINT engine
SITE ADMINISTRATION

Welcome, J. Archivist. [Log out]

Home > Site Configuration

About Your Site

- Edit descriptive texts for this site
- Change this site's name
- Define the name for items in your collection
- Edit the featured text
- Set your metadata preferences

Users

- Manage user profiles
- Edit user info & powers

Type Management

- Create categories, descriptions and document types for images, texts and media files

Search/Browse Options

- Decide how users can browse and search your site

Community Tools

- Enable/disable visitor commenting
- Delete comments

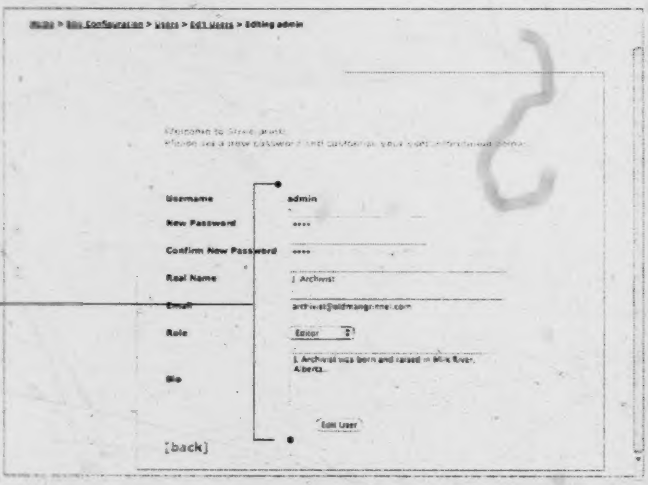
Course/Syllabus Tools

- Create courses and syllabi
- Add items from your collection to syllabi
- Upload files and content for courses

[back]

Users

Fig 12. Editing the admin user



Home > Site Configuration > Users > Edit Users > Editing admin

Username: admin

New Password: ****

Confirm New Password: ****

Real Name: J. Archivist

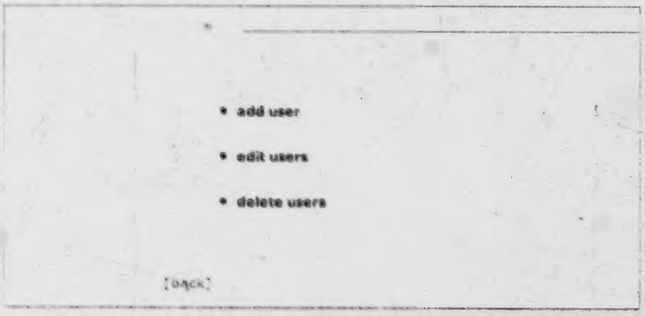
Email

Role: Editor

Bio

The Users section will be your first encounter with Streetprint's administrative interface. The first time you log in, you'll be prompted to change your password and enter some other information:

If you have more than one person working on your Streetprint site, you can add user accounts for them very easily.



- add user
- edit users
- delete users

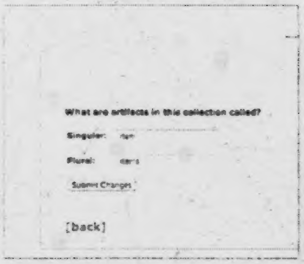
Fig 13. The Users screen

Not all accounts are created equal, however. Editors are users who have the same privileges as your admin user: they can configure the site and add/edit/delete items to your collection. Data entry users can only add collection items and edit records which they added themselves. This helps you maintain editorial control over your new Streetprint site if you so choose.

Later on, we'll show you how the information you enter for your user accounts can be displayed automatically on your Streetprint site's "About" page.

About Your Site

The About Your Site section lets you configure most of the things visitors will first notice when visiting your online Streetprint archive.



What are artifacts in this collection called?

Singular: item

Plural: items

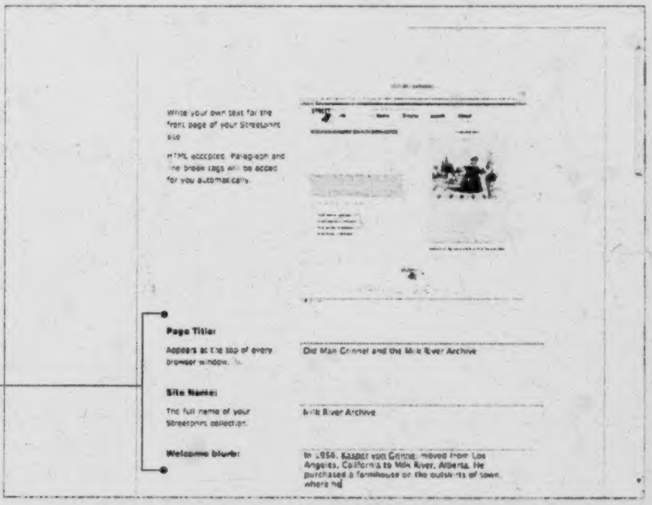
Fig 14. That's one hot item!

Artifact Name: The Artifact Name screen allows you to change the word used to describe items in your collection. "Texts" is the default setting. Many pieces in Old Man Grinnel's eccentric collection—outside academic circles, at least—wouldn't normally be considered "texts." Borrowing the nomenclature we've been using in this manual, let's change the artifact name from "texts" to "items:"

Front & About Pages: This section lets us change the text blurbs which appear in various places around our website, as well as the name of our site itself. The page is divided into three sections, each with its own button for submitting changes. The first section lets us set our *page title* and *site name.*

In our case, we've opted to have "Old Man Grinnel and the Milk River Archive" appear at the top of our website's browser window *(the page title)*, and use the shorter "Milk River Archive" as the *site name.*

Fig 15. Editing the front page and site names



Write your own text for the front page of your Streetprint site

HTML accepted. Paragraph and line break tags will be added for you automatically.

Page Title:

Appears at the top of every browser window.

Old Man Grinnel and the Milk River Archive

Site Name:

The full name of your Streetprint collection.

Milk River Archive

Welcome blurb:

In 1956, Kasper von Grinnel moved from Los Angeles, California to Milk River, Alberta. He purchased a farmhouse on the outskirts of town, where he

The rest of this section allows you to specify the text which appears on the website's home page, on the site's "About" page, and in the fine print at the bottom of every page. It also provides a field for describing the equipment and methodology used for digitizing the items in your site, and even a checkbox for displaying information about your website's contributors on the site's "About" page. When checked, the "About" page will display the name, email address, and biography of every user that has been set up in the *Users* section, listing them under the heading "Project Staff."

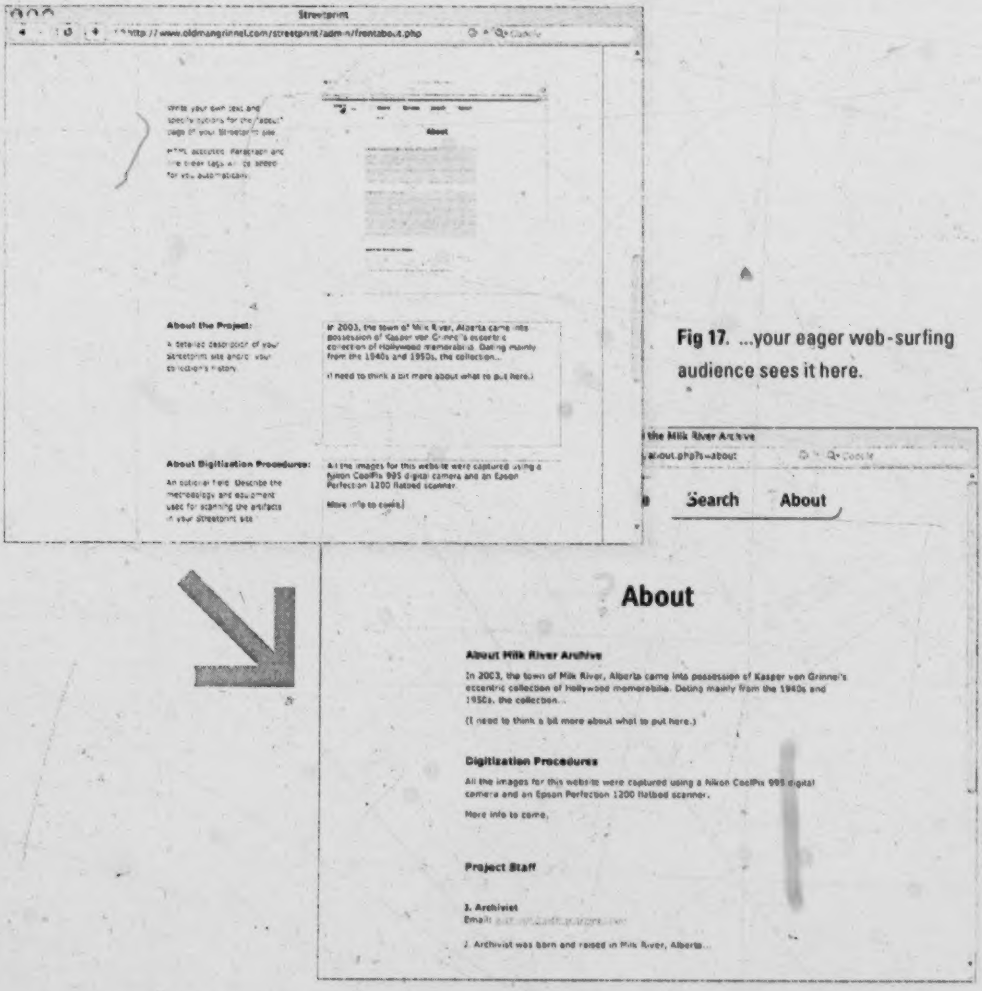


Fig 16. When you type this...

Streetprint
http://www.oldmangrinnel.com/streetprint/admin/frontabout.php
About

Write your own text and specify options for the "about" page of your Streetprint site.

HTML accepted. Paragraph and line break tags will be added for you automatically.

About the Project:

A detailed description of your Streetprint site and/or your collection's history.

In 2003, the town of Milk River, Alberta came into possession of Kasper von Grinnel's eccentric collection of Hollywood memorabilia. Dating mainly from the 1940s and 1950s, the collection...

(I need to think a bit more about what to put here.)

About Digitization Procedures:

An optional field. Describe the methodology and equipment used for scanning the artifacts in your Streetprint site.

All the images for this website were captured using a Nikon CoolPix 995 digital camera and an Epson Perfection 1200 flatbed scanner.

More info to come.)

Fig 17. ...your eager web-surfing audience sees it here.

About

About Milk River Archive

In 2003, the town of Milk River, Alberta came into possession of Kasper von Grinnel's eccentric collection of Hollywood memorabilia. Dating mainly from the 1940s and 1950s, the collection...

(I need to think a bit more about what to put here.)

Digitization Procedures

All the images for this website were captured using a Nikon CoolPix 995 digital camera and an Epson Perfection 1200 flatbed scanner.

More info to come.

Project Staff

J. Archivist
Email:

J. Archivist was born and raised in Milk River, Alberta...

For Power Users:
Digitization Procedures

Your "Digitization Procedures" text field is the perfect place to enter precise technical information on your archival process once it has been established. As an example, here's the Digitization Procedures text from Streetprint's flagship collection, Revolution and Romanticism:

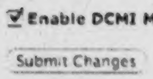
> **All images on "Streetprint: Revolution and Romanticism" were catalogued between July 2002 and March 2003 by Chris Govias.**
>
> **Images were scanned as 300 dpi TIFF files using an Epson Perfection 1250 flatbed scanner. Detailed images (such as woodcuts) were captured at 600 dpi using the same equipment. Oversized images were captured with a Nikon CoolPix 995 3.34 megapixel digital camera.**
>
> **These raw images were then resized, color-corrected, and optimized for the web using Adobe Photoshop 7. A series of Photoshop actions automated this procedure. The images were then added to the site using the Streetprint Engine's image management system (Version 1.0).**

Featured Text: Every Streetprint site has a featured text, an artifact or item which is displayed on the front page of your website. This is a great way to draw visitors into your collection, provide some dynamic content, or just show off your favourite image—you can change the featured item as often as you like.

Once your featured item is selected, you can optionally select an image other than the default image to display; this could be useful for highlighting an unusual feature inside a text instead of displaying the cover, for example.

Since we don't have any items in our collection yet, we'll come back here later...

Metadata: The Metadata screen currently allows you to do one simple thing: turn your Streetprint's site's metadata on and off. (It is on by default.)



☑ Enable DCMI Metadata

Fig 18. How to enable Metadata

What is metadata? Simply put, metadata is "data about data." While Old Man Grinnel's collection is definitely unique, it is useful to provide ways for other organizations and databases to be able to "read" our item information in a common format.

Streetprint's current implementation of this feature provides basic metadata using the Dublin Core Metadata Initiative's metadata terms. If the "Enable DCMI Metadata" box is checked, DCMI-compliant metadata tags will be added to the source code of your Streetprint site's item/text pages. While this change will not be visible to most browsers, any software which is "metadata aware" can use this additional information.

The metadata tags added to your Streetprint site's text/item pages will look something like this:

Fig 19. Metadata tags

```
<meta name="DC.Title" content="Letter from Ed Nagy">
<link rel="SCHEMA.dc" href="http://purl.org/metadata/dublin_core_elements#title">

<meta name="DC.Creator" content="Edward Nagy">
<link rel="SCHEMA.dc" href="http://purl.org/metadata/dublin_core_elements#creator">

(etc.)
```

Team Streetprint welcomes feedback on our metadata implementation from experts in this field. The Streetprint Engine's architecture permits the addition of multiple metadata formats and XML export options in the future, should this become desirable.

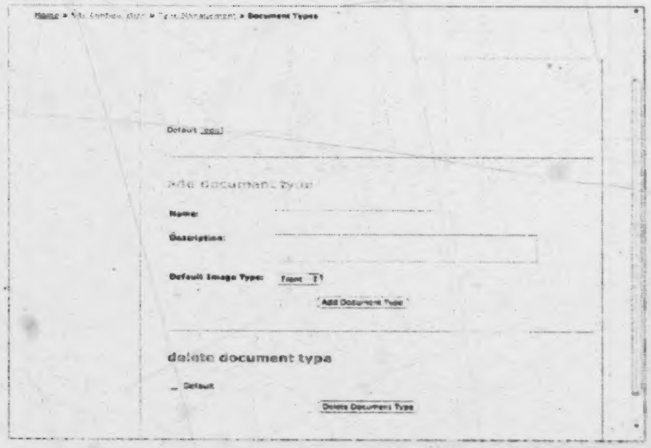
* For more information on Dublin Core Metadata, visit *www.dublincore.org*.

Type Management

Every collection is different, and so the Streetprint Engine's *Type Management* section lets you organize your collection in a useful way, creating a research taxonomy which will be unique to your site. As we'll see later, this will also shape the ways in which users will be able to search and browse your Streetprint site.

The four different "types" which this section lets you manage are *document types, categories, image types, and media types.* We'll examine them in order.

Document Type: Every text, item, or artifact in your Streetprint site has a single document type. In Streetprint lingo, a document type is simply the name for the kind of physical object in question; it's the answer to the question "so what is this thing?". Every item in your collection must have one (and only one) document type.


Home > [illegible] > [illegible] > Document Types
Default edit
add document type
Name:
Description:
Default Image Type:
Add Document Type
delete document type
Delete Document Type

Default Image Type: Front

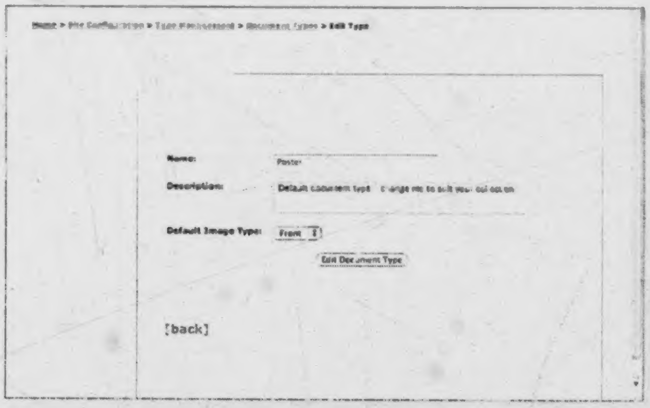
☐ Default

Fig 20. Managing document types

When you click on "document types" from the Type Management screen, you will see an interface which is shared with all the types in this section, a single screen from which you can edit current types, add new types, or delete existing types.

Many of the items in Old Man Grinnel's collection are movie posters, so let's make "Poster" our first document type. You'll notice that we already have a single document type, named "Default." Instead of creating a whole new document type, let's edit this Default type (by clicking the 'edit' link next to its name) and rename it to "Poster."

Fig 21. Defaults? We don't need no stinkin' defaults!


Home > Site Configuration > Type Management > Document Types > Edit Type
Edit Document Type
[back]

Name: Poster

Description: Default document type - change me to suit your collection

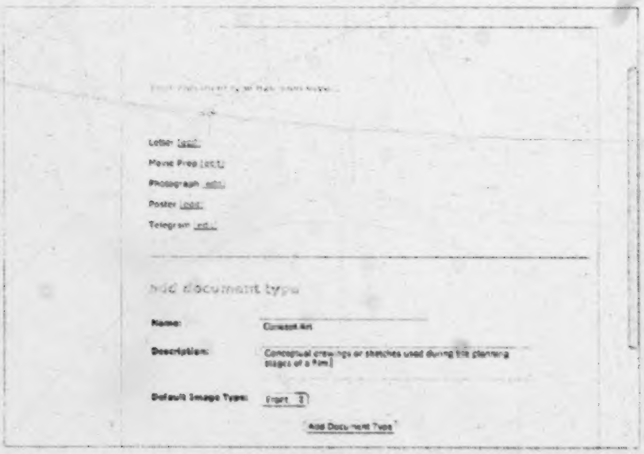
Default Image Type: Front

The "description" field (for all types in the 'type management' section) is optional, and can be left blank. It might be useful, however, to use this field to document any notes you have

about the type you are setting up, especially if you are working on your site with other collaborators. Future versions of the Streetprint Engine may also use the information in these fields in an expanded version of the "About" section.

What other document types do we need? After sifting through all the boxes in Old Man Grinnel's collection, we've discovered the following kinds of items:

- Photograph
- Concept Art
- Movie Prop
- Letter
- Telegram



Letter [edit]
Movie Prop [edit]
Photograph [edit]
Poster [edit]
Telegram [edit]

Add document type

Name: Concept Art

Description: Conceptual drawings or sketches used during the planning stages of a film

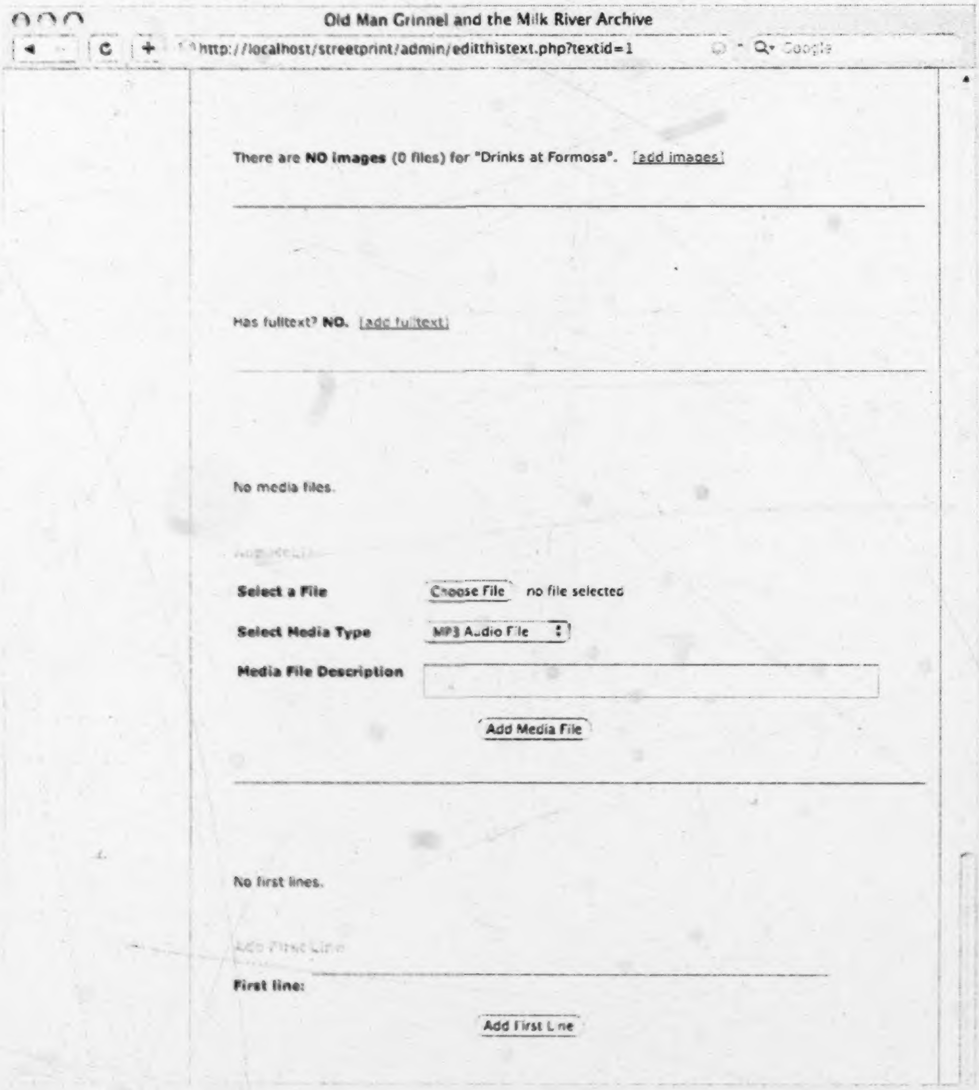
Default Image Type: Front

We can use the document types screen to add these types. One detail to note is the "Default Image Type" pull-down menu, which will be set to "Front" by default. We can safely ignore it for now; this menu's function will be explained later in this section when Image Types are discussed.

Fig 22. Adding more document types

For Power Users:
Deleting Types

Wait a second, what happens if I have 30 "books" in my collection and then I delete the document type "book"? Don't worry, Streetprint won't let you do this. You'll get a warning message, and you'll have to delete (or pick new document types for) all the individual books in your collection before you can get rid of the document type. This probably won't occur very often, however; most of the time, you'll simply want to rename your document types (or categories) instead of deleting them.



Old Man Grinnel and the Milk River Archive

http://localhost/streetprint/admin/editthistext.php?textid=1

There are **NO images** (0 files) for "Drinks at Formosa". [add images]

Has fulltext? **NO.** [add fulltext]

No media files.

Add Media

Select a File no file selected

Select Media Type MP3 Audio File

Media File Description

No first lines.

Add First Line

First line:

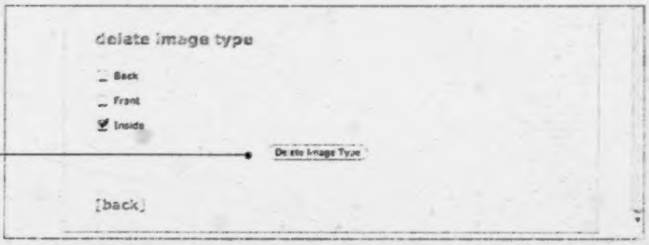
Fig 37. The second half of Streetprint's "Edit Details" page.

Categories: Categories are similar to Document Types, but with one key difference: items in your Streetprint collection can belong to multiple categories, whereas each item can only have one document type. Because of this, the uses for the Categories type are nearly endless; you can use them to set up thematic categories, use them as keywords, or set up any other sort of distinctions you like.

One thing the document types we've just set up for Old Man Grinnel's collection don't tell us is what kind of subject matter the artifacts address. Categories can help us address this problem, and we can always add more as we need them. For now, we've come up with the following categories for our archive:

- Science Fiction
- Horror
- Thriller
- Comedy
- Actors
- Directors
- Los Angeles
- Milk River
- Thinly Veiled Communist Allegories
- (Etc.)

Image Types: Image types are like document types, but for images. Every image you add to your Streetprint site must have one (and only one) image type. By default, Streetprint comes with three image types that clearly have books in mind: Front (for front covers), Back (for back covers), and Inside (for inside pages). Since Old Man Grinnel's collection consists mainly of artifacts such as posters and photographs, "Front" still makes sense as an image type, and we could perhaps keep "Back" in there as well, in case there is some writing (or lipstick) on the back of a document and we would like to include it in the site. "Inside" doesn't really apply to our collection, however, so let' delete it.



delete image type

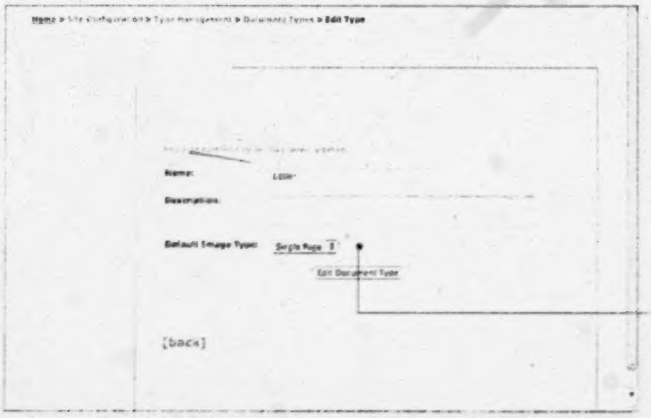
☐ Back
☐ Front
☑ Inside

Fig 23. Deleting an image type

Instead, we've come up with the following additional image types:

- Single Page (for letters, which we will scan one page at a time)
- Close-up (for artifacts where we want to provide additional close-up images highlighting certain details)

You might remember that when creating new Document Types, we saw a pull-down menu named "Default Image Type." One of the nice features of a Streetprint site is that, when browsing, a user sees a small "thumbnail image" above the name of items in the collection, letting them see the artifact instead of just reading its name. If an artifact has several images, however—let's say it's a movie prop and it has a "Front" image (a photograph of the whole prop) and three "Close-up" images (photos of the prop from different angles)—how does Streetprint know which single thumbnail image it should choose to represent the artifact as a whole? The answer is that it looks at the document type of the artifact in question, and then chooses the first image which matches the document type's "Default Image Type."



Home > Site Configuration > Type Management > Document Types > Edit Type

Name: Letter

Description:

Default Image Type: Single Page

For our collection, having the default image type set to "Front" is usually a good choice. There's one exception, however: we decided earlier that we would use the "Single Page" image type for all the pages of a letter, including the first page. This means we should set Letter's default image type to "Single Page":

Fig 24. Please, Mr. Postman: change my default image type

Every time we add images to our Streetprint site, we'll have the opportunity to select the appropriate image type. We'll show you how in the next chapter of this manual.

Media Types: One of Streetprint's more advanced features is the ability to add a variety of media files to the record of any item in your collection. If an artifact has media files associated with it, a user browsing your site will see a "Media" button. When clicked, this button will list the artifact's media and provide download links.

As you've probably guessed, Media types are a lot like Document or Image types. Streetprint knows about two media types by default: MP3 audio files and Quicktime Movies. Adding media types is extremely easy; all you need is a name, a description, and the file "extension" belonging to files of this kind. ("Extension" simply refers to the letters or numbers which occur after the "." in the filename—mp3, doc, mpg, etc.)

For Power Users:

Uses for Media Types

Media types are a great way to associate any kind of file with items in your collection. Simply by adding the appropriate media type, you can incorporate everything from PDFs and Powerpoint files to Flash movies into your Streetprint site. A current shortcoming, however, is that Streetprint simply provides download links in this format, regardless of media type:

> **-- testmovie.mov (Quicktime Movie, 1.2 MB)**
> **This movie is a test movie.**

There currently is no default way to have some content (movies or sound files) play "inline."

Important note: By default, most PHP servers limit uploaded file sizes to 2MB or smaller. If you plan to add media files which are larger than 2MB, you may need to contact your server administrator and have them change this setting first.

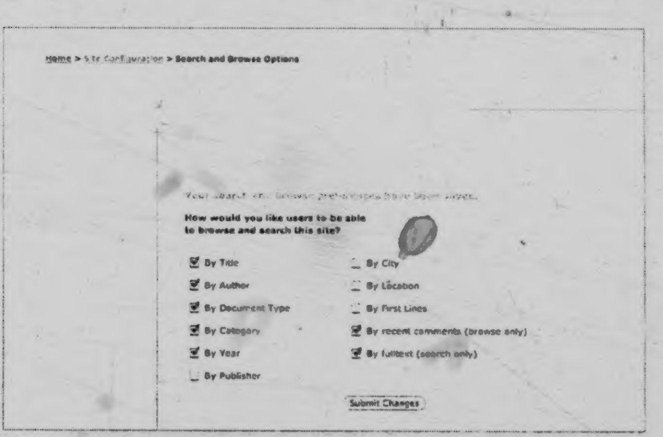
Now that we've set up all of our site's types, we're nearly ready to add items to our collection, the topic of the next chapter. But rather than cut right to the chase like a bad Hollywood B-movie, let's develop our plot a bit more and take a look at the rest of the 'Site Configuration' section...

Search / Browse Options

This section consists of a single, easy-to-use screen. Because every artifact in our Streetprint site will have a title, a document type, and many other optional fields of information, we need to let the Streetprint Engine know which of these are important for users browsing and searching our site.

For the Milk River Archive, we likely won't be using the "publisher" field very often, we don't have any "first lines" (a field mostly useful for poetry), and fields such as "Location" and "City" aren't of primary importance. We've decided to chose the following search and browse options:

We'll examine most of these fields at greater length in the next chapter.



Home > Site Configuration > Search and Browse Options

Your search and browse preferences have been saved.

How would you like users to be able to browse and search this site?

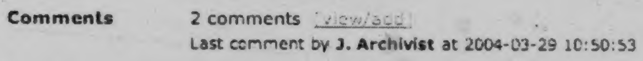
☑ By Title
☑ By Author
☑ By Document Type
☑ By Category
☑ By Year
☐ By Publisher
☐ By City
☐ By Location
☐ By First Lines
☑ By recent comments (browse only)
☑ By fulltext (search only)

Fig 25. Editing search and browse options.

Community Tools

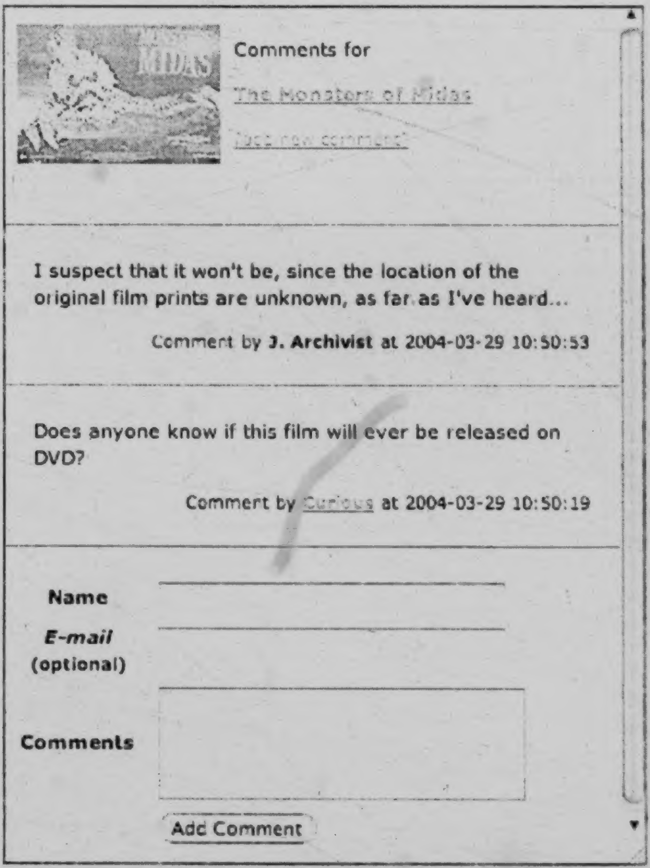
Team Streetprint plans to gradually build more and more "community tools" into the Streetprint Engine. The Community Tools section is new in version 2.1, and is currently used for managing *user comments*.

By default, every Streetprint site features the ability for site visitors to add their own comments to the texts/items in your digital archive. Visitors to your site will see comments displayed like this:



Comments 2 comments [view/add]
Last comment by **J. Archivist** at 2004-03-29 10:50:53

Fig 26. Everybody's talkin at me... A 'comments' entry when viewing a collection item with two comments.



Comments for

The Monsters of Midas

[add new comment]

I suspect that it won't be, since the location of the original film prints are unknown, as far as I've heard...

Comment by **J. Archivist** at 2004-03-29 10:50:53

Does anyone know if this film will ever be released on DVD?

Comment by Curious at 2004-03-29 10:50:19

Fig 27. Clicking on a "view" or "add" comment link displays a popup window like this one.

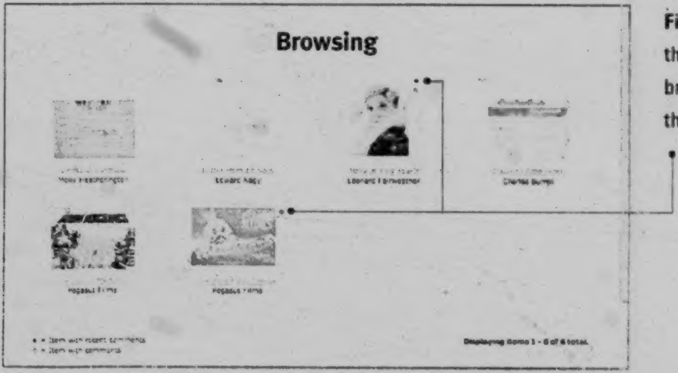
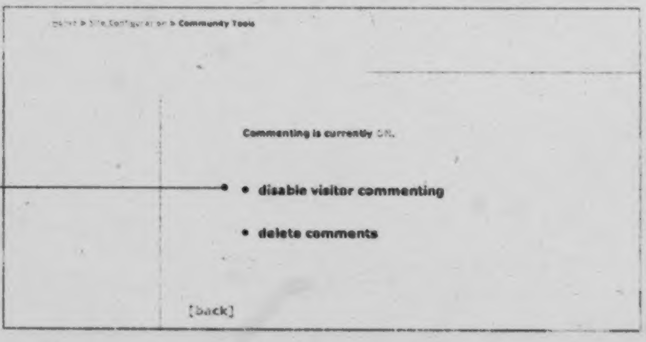


Fig 28. Small symbols indicate the presence of comments while browsing. Clicking them will bring up the appropriate pop-up window.

If you would prefer not to feature user comments on your Streetprint site, you can turn this feature off with a single click on the "Community Tools" page. It can also be re-enabled at a later date if you so choose.



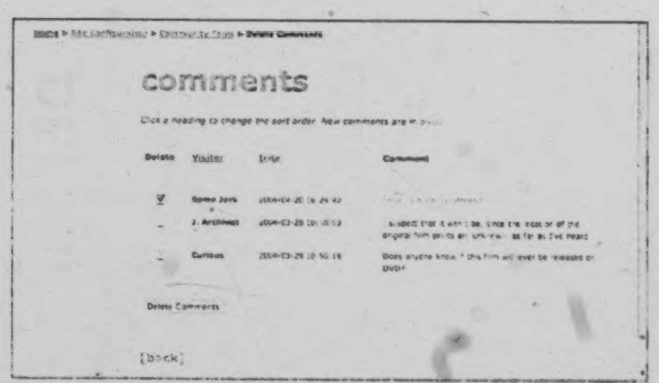
[illegible] > Site Configuration > Community Tools

Commenting is currently ON.

- disable visitor commenting
- delete comments

[back]

Fig 29. Click here to disable user comments



[illegible] > Site Configuration > Community Tools > Delete Comments

comments

Click a heading to change the sort order. New comments are in blue.

Delete	Visitor	Date	Comment
☑	[illegible]	2004-04-20 [illegible]	[illegible]
☐	J. Archivist	2004-03-29 [illegible]	I suspect that it won't be, since the location of the original film prints are unknown, as far as I've heard
☐	Curious	2004-03-29 10:50:19	Does anyone know if this film will ever be released on DVD?

Delete Comments

[back]

Fig 30. Deleting Comments.

The Streetprint Engine makes it easy to keep an eye on your user comments and delete inappropriate comments as necessary. Clicking on "Delete Comments" from the main Community Tools page will take you to a screen listing all your site's comments in reverse-chronological order. Comments which are new will appear in blue.

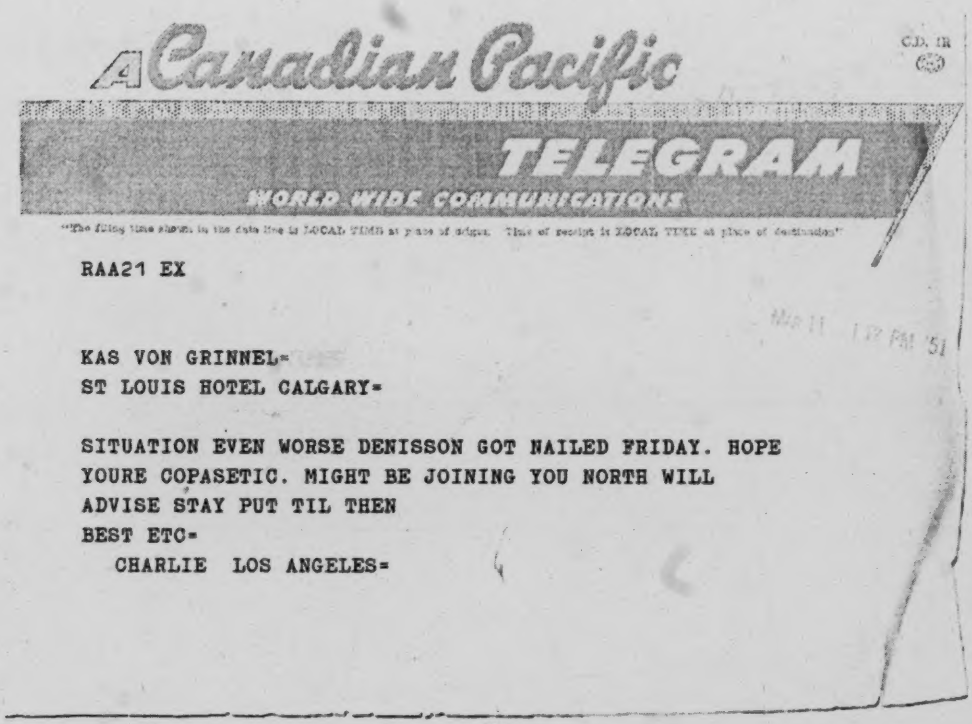
To delete a comment, simply check the box next to it and click the "Delete Comments" button. Use this tool sparingly; once comments have been deleted, they cannot be recovered.

Course / Syllabus Tools

This section of Streetprint s administrative interface is aimed at instructors using Streetprint as an instructional aid. It will be documented in a later version of this user manual. If you want to use Streetprint in your teaching, however, give it a try!

CHAPTER IV

MANAGING YOUR COLLECTION



A Canadian Pacific

C.D. IR

TELEGRAM

WORLD WIDE COMMUNICATIONS

"The filing time shown in the date line is LOCAL TIME at place of origin. Time of receipt is LOCAL TIME at place of destination"

RAA21 EX

Mar 11 1 12 PM '51

KAS VON GRINNEL=
ST LOUIS HOTEL CALGARY=

SITUATION EVEN WORSE DENISSON GOT NAILED FRIDAY. HOPE YOURE COPASETIC. MIGHT BE JOINING YOU NORTH WILL ADVISE STAY PUT TIL THEN
BEST ETC=
CHARLIE LOS ANGELES=

Fig 31. "Situation Even Worse," Telegram to Kasper von Grinnel from Charles Burrell, 1951

Overview

Now that our Streetprint site for the Milk River archive has been installed and configured with a site name, document types, categories, and more, it's time to begin adding items to our collection!

The process of adding a collection item to your Streetprint site has five steps, and generally proceeds in this order:

Adding to Your Collection – Steps:

1. Image capture (scanning or taking a digital photograph of the item) *p* 35
2. Image processing: enhancing the image if neccessary and re-sizing it for the web. *p* 36
3. Adding a Streetprint collection record for the item. *p* 42

4. Adding images to the collection record *p 48*
5. Optionally adding media files, fulltext, etc. to the collection record *p 47*

These steps don't need to proceed sequentially for every individual collection item, however. If it is more convenient, for example, you could complete the first two steps (image capture and processing) for every item in your collection before adding any collection records to your Streetprint site (step three). Likewise, each step can happen at a different time, or be performed by a different person if you so choose.

About Images: It becomes apparent from the list of steps above that *images* are extremely important to a Streetprint site. This stems from our roots in the study of street literature, and our belief that with most artifacts—textual or otherwise—recreating the experience of sitting down with the physical item itself is of utmost importance. This focus on image-based representation (over fulltext, tagging, or other kinds of presentation methods) is perhaps Streetprint's defining feature. This focus does have a downside, however: the scanning and processing of artifact images is by far the most labour-intensive aspect of creating a Streetprint collection, and these activites remain (for now) external to Streetprint. This means that the current version of the Streetprint Engine cannot do scan-ning and image editing for you. Instead, we recommend using third-party applications such as Adobe® Photoshop®*, The Gimp**, and/or the software that came with your digital camera or scanner.

This chapter will focus primarily on the collection management steps which directly involve the Streetprint Engine. We will also provide a brief overview of how we manage images for our own Streetprint sites in the CRC Studio.

* See *www.adobe.com/products/photoshop/* for more information.

** See *www.gimp.org* for more information.

Garbage In, Garbage Out

"Garbage in, garbage out" is a frequently-used expression in the computer world, and one you should keep in mind as you begin adding items to your Streetprint collection. The phrase conveys an unfortunate truth about computer software and databases: they're only as good as the information you put into them. The best database in the world will be useless if the data in it is inconsistent; imagine trying to search a library card catalogue that contained twelve different spellings of "Shakespeare," for example.

Software developers sometimes solve the "garbage in, garbage out" problem by making programs that are extremely rigid and inflexible, alerting the user to every possible error in input. This is possible (and even desireable) in some specialized cases, but what about a system like the Streetprint Engine which is designed to be as flexible as possible? Though many aspects of the current Streetprint Engine betray our literary roots, we hope that it will be increasingly used for a wide array of archives and collections about which we have little specific knowledge. For this reason, we can't make the Streetprint's error-checking *too* stringent for fear of limiting the flexibility and usefulness of the software. Instead, our design philosophy presumes an intelligent site editor. We don't think this is an unreasonable decision...after all, it's your collection!

What does this mean for you? Avoiding the "garbage in, garbage out" problem is easy if you set *standards and procedures* for your Streetprint site when you begin adding to your collection. If you are working with other collaborators—or even if you're not—write down the decisions you're making as you scan your first few im-ages and add your first few text records, and to use these as a basis for formalized procedures. Are authors listed in "firstname lastname" format, or "lastname, firstname?" Should artifacts be scanned at 300 or 600 dpi? What kind of information should go in the "location" field, and how should it be formatted? While the rest of this chapter will make suggestions and provide the background you need to make decisions such as these, it's ultimately *your* call. The most important thing is that once choices like this are *made*—even if they're relatively arbitrary—all subsequent additions to your Streetprint site should be consistent and adhere as closely as possible to your standards. If this basic principle is followed, your Streetprint site can avoid "garbage in, garbage out" syndrome and continue to be a useful resource well into the future!

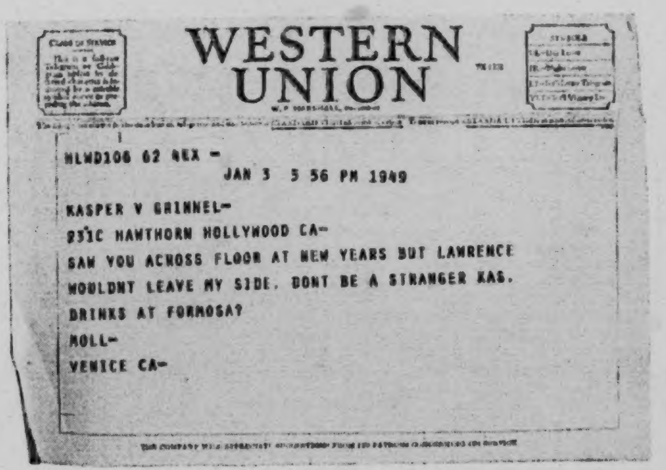
So how do we add collection items to a Streetprint site, anyway? Let's start at the very beginning...

Image Capture

One of the first steps when adding to your Streetprint collection is "image capture," the process of creating digital images of the items in your archive. There are as many ways to create digital images as there are kinds of artifacts, and it is impossible to describe them all here. For the purposes of this manual and our Milk River archive, we're going to assume that our equipment consists of a consumer-grade flatbed scanner (readily available for under $200) and a high-end consumer-level digital camera.

If you work or study at a university, it is worth investigating the digitization resources which may be offered by various groups on your campus. Often places such as technical services, an art and design department, or science laboratories will have the equipment and/or facilities neccessary to digitize items which (by virtue of their size or shape) are difficult to capture with more modest resources.

For small and flat collection items, such as books or paper, capturing images with a flatbad scanner usually provide the best combination of ease-of-use and image quality. Here is a sample telegram from Old Man Grinnel's collection, captured using a flatbed scanner:



WESTERN UNION

NLWD106 62 NEX -
JAN 3 5 56 PM 1949

KASPER V GRINNEL-
931C HAWTHORN HOLLYWOOD CA-
SAW YOU ACROSS FLOOR AT NEW YEARS BUT LAWRENCE
WOULDNT LEAVE MY SIDE. DONT BE A STRANGER KAS.
DRINKS AT FORMOSA?
MOLL-
VENICE CA-

Fig 32. Telegram from Molly Heatherington to Kasper von Grinnel, 3 Jan 1949

This telegram is a ideal example of an item well-suited to flatbed scanning; even the shadows cast beneath the page's folded edge contribute to a relatively accurate representation of the object. This telegram was scanned at 300 DPI (dots per inch), an image resolution which permits both online distribution (such as through your Streetprint site) and print distribution. While this resolution is likely higher than neccessary for Streetprint purposes alone, the ability to later produce a print version (or a higher-resolution web version) without having to re-scan the original artifact is a definite asset. If you anticipate any future need for extremely high-resolution versions of your collection items, or intend to use your scans as definitive archival copies, scan at resolutions higher than 300 DPI.

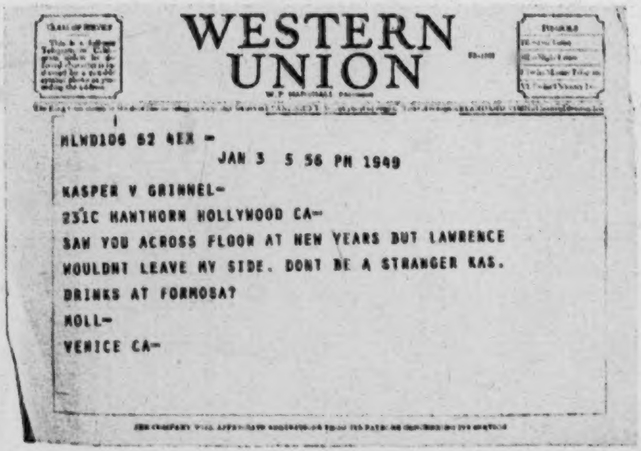
For books whose spines cannot be broken, oversize items (such as Old Man Grinnel's movie posters), or more three-dimensional objects, digital photography may be a better option than flatbed scanning. Photography is an art unto itself, but a few tips can greatly improve your capture results: use your camera's highest-resolution setting, ensure a white or neutral background with uniform lighting, use a tripod, and consult a friend/colleague/student with photographic training if at all possible. With proper digital photography and a bit of image editing, digital photos are often indistinguishable from flatbed scans when viewed at web resolutions.

Image Processing

As we mentioned in the introduction to this chapter, image processing can be accomplished with a variety of different software programs. Since the Streetprint Engine is not directly involved in this process, this section will simply mention a few tips and tricks, as well as describe ideal image sizes and resolutions for use with your Streetprint site.

Correcting Image Capture Problems: The scan we made of Kasper von Grinnel's telegram was good, but it had a few minor problems: the telegram is slightly crooked, it looks a bit more faded than the original, and there is too much white space around the edges of the artifact. Using Adobe® Photoshop®, we can correct these problems using the Rotate function, the Brightest/Contrast controls, and the crop tool, respectively. We can then save the resulting file as a TIFF image and add something like 'Fixed Scan' to the end of its name. The resulting image looks like this:

Fig 33. Telegram from Molly Heatherington to Kasper von Grinnel, after initial image processing

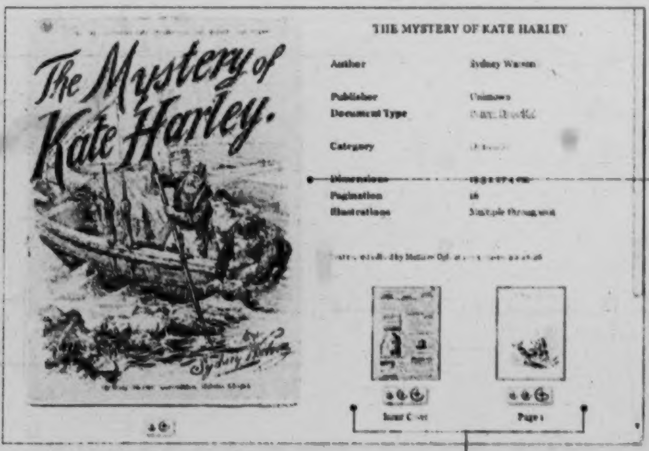


WESTERN UNION

MLWD106 62 4EX -
JAN 3 5 56 PM 1949

KASPER V GRINNEL-
231C HAWTHORN HOLLYWOOD CA-
SAW YOU ACROSS FLOOR AT NEW YEARS BUT LAWRENCE
WOULDNT LEAVE MY SIDE. DONT BE A STRANGER KAS.
DRINKS AT FORMOSA?
MOLL-
VENICE CA-

Making Resolutions: Now that we have a digital image which properly represents the actual artifact, it's time to prepare it for inclusion in our Streetprint site. Our current file, a 300 dpi image in TIFF format, is much too large (in terms of both file size and pixel size) to put online directly. Instead, we need to prepare a few differently-sized versions—called "resolutions" in Streetprint lingo—of the image. *At minimum, every image you wish to include on your Streetprint site requires two resolutions*, a "Thumbnail" image (miniturized representation of the image) and a "Low" resolution image. Examples of these two image resolutions can be seen in the illustration below, taken from Revolution and Romanticism:



THE MYSTERY OF KATE HARLEY

Author Sydney Watson
Publisher Unknown
Document Type [illegible]
Category [illegible]
Dimensions [illegible]
Pagination 16
Illustrations Multiple throughout

"Low" resolution image

"Thumbnail" resolution images

Fig 34. Image resolutions from www.crcstudio.arts.ualberta.ca/streetprint/

Oftentimes the "Low" resolution image won't show enough detail, and so the Streetprint Engine can optionally accomodate two additional resolutions, "Medium" and "High." By providing 3 (or even 4) different resolutions for each image on your Streetprint site, you can offer your visitors maximum flexibility and legibility, allowing those with fast connections to browse very large images and visitors with dial-up connections to view smaller (but faster-loading) versions of the same image. (The plus signs visible underneath the thumbnails in the previous illustration allow visitors to select which resolution they would like to use.)

So Many Sizes: So, how do you know exactly how big (or small) to make your images? In theory, Streetprint will handle any size of image for any resolution, but the layout of a Streetprint site won't work too well if, say, your thumbnail images are 600 pixels wide. Here are the guidelines that Team Streetprint uses when preparing images for our own sites; we recommend you stay fairly close to these parameters for the best results with your Streetprint site.

Resolutions	**Suggested Dimensions**	**Format**	**Ideal file size***
Thumbnail	120 x 120 pixels or less	JPEG	6 KB or less
Low	500 x 500 pixels or less	JPEG	30 – 60 KB or less
Medium	800 – 1000 pixels or less	JPEG	100 – 200 KB or less
High	Larger than 1000 pixels squared	JPEG	More than 200 KB

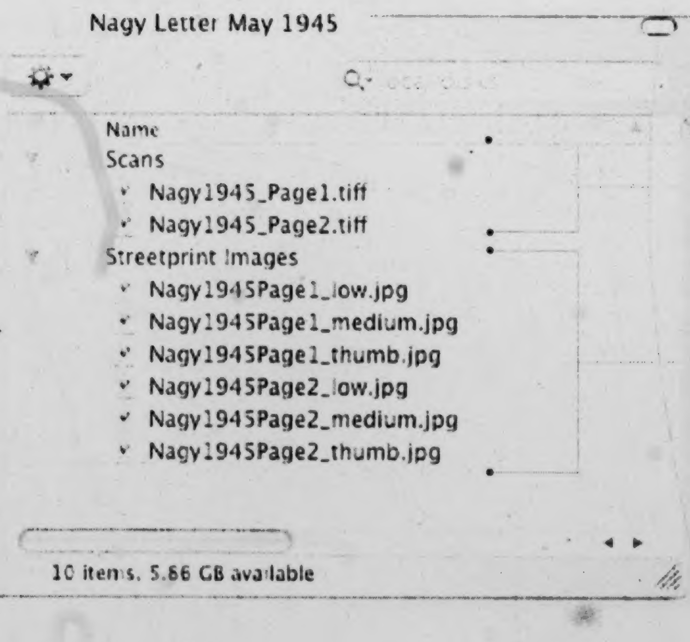
Fig 35. Streetprint Image Resolution Guidelines

*recommendation only, it may not always be possible to achieve these sizes

Once you have used image editing software to scale your original scan down to one of the sizes listed above, you should finish by saving the new file with your software's "Save for Web" function, if available. The Streetprint Engine currently expects that all images will be in JPEG format, although its architecture supports additional image formats should new standards emerge. (This is another good reason to always keep a copy of your original TIFF scans; since the format is uncompressed, you can return to these files in future years to produce versions of them in new image formats.)

Does all this talk of multiple sizes, resolutions and files make your head spin? In the CRC Studio, Team Streetprint has come up with a good method of keeping our image files organized, making it a snap when we later upload the images to our Streetprint site.

Let's say we're scanning a two-page letter from Old Man Grinnel's archive. We would arrange our computer's files and folders like this:



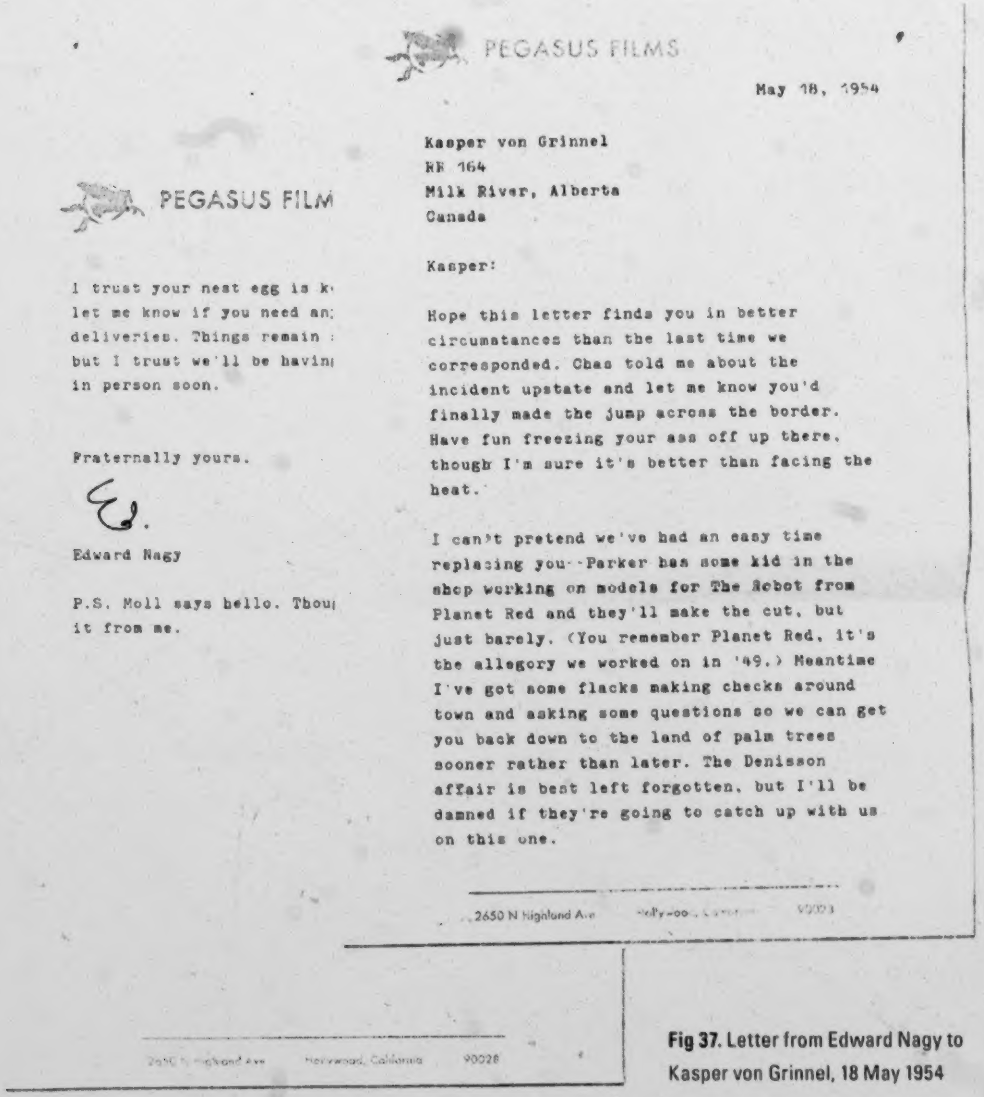
Nagy Letter May 1945

Name
Scans
Nagy1945_Page1.tiff
Nagy1945_Page2.tiff
Streetprint Images
Nagy1945Page1_low.jpg
Nagy1945Page1_medium.jpg
Nagy1945Page1_thumb.jpg
Nagy1945Page2_low.jpg
Nagy1945Page2_medium.jpg
Nagy1945Page2_thumb.jpg

10 items, 5.66 GB available

- Images folder for 'Nagy Letter 1945'
- Original scans
- Web-ready images

Fig 36. Image Organization



PEGASUS FILMS

May 18, 1954

Kasper von Grinnel
RR 164
Milk River, Alberta
Canada

Kasper:

Hope this letter finds you in better circumstances than the last time we corresponded. Chas told me about the incident upstate and let me know you'd finally made the jump across the border. Have fun freezing your ass off up there, though I'm sure it's better than facing the heat.

I can't pretend we've had an easy time replacing you--Parker has some kid in the shop working on models for The Robot from Planet Red and they'll make the cut, but just barely. (You remember Planet Red, it's the allegory we worked on in '49.) Meantime I've got some flacks making checks around town and asking some questions so we can get you back down to the land of palm trees sooner rather than later. The Denisson affair is best left forgotten, but I'll be damned if they're going to catch up with us on this one.

2650 N Highland Ave Hollywood, California 90028

PEGASUS FILM

I trust your nest egg is k
let me know if you need an
deliveries. Things remain
but I trust we'll be havin
in person soon.

Fraternally yours,

E.

Edward Nagy

P.S. Moll says hello. Thou
it from me.

2650 N Highland Ave Hollywood, California 90028

Fig 37. Letter from Edward Nagy to Kasper von Grinnel, 18 May 1954

For Power Users:

Streetprint Actions for Adobe® Photoshop®

One of the many useful features of Adobe® Photoshop® image editing software is the ability to create "actions," a series of pre-recorded steps which can then be executed with a single click. The process of creating multiple Streetprint resolutions from each and every scan is admittedly repetitive and laborious; actions can help reduce the tedium and speed up the process considerably.

While we encourage you to create your own set of actions—consult your Adobe® Photoshop® manual to find out how—Team Streetprint has also made its own actions available for you to download and use. They can be found online at:

www.streetprint.org/engine_downloads.php

Double-clicking the StreetprintActions.atn file will load six Streetprint actions into your Photoshop "Actions" tab. They provide automatic resizing and sharpening for two kinds of document orientation—Portrait (tall) and Landscape (wide)—and three Streetprint resolutions (thumbnail, low, and medium). Once an action has been applied to a source TIFF image, you can "Save for Web..." and your image is ready to go!

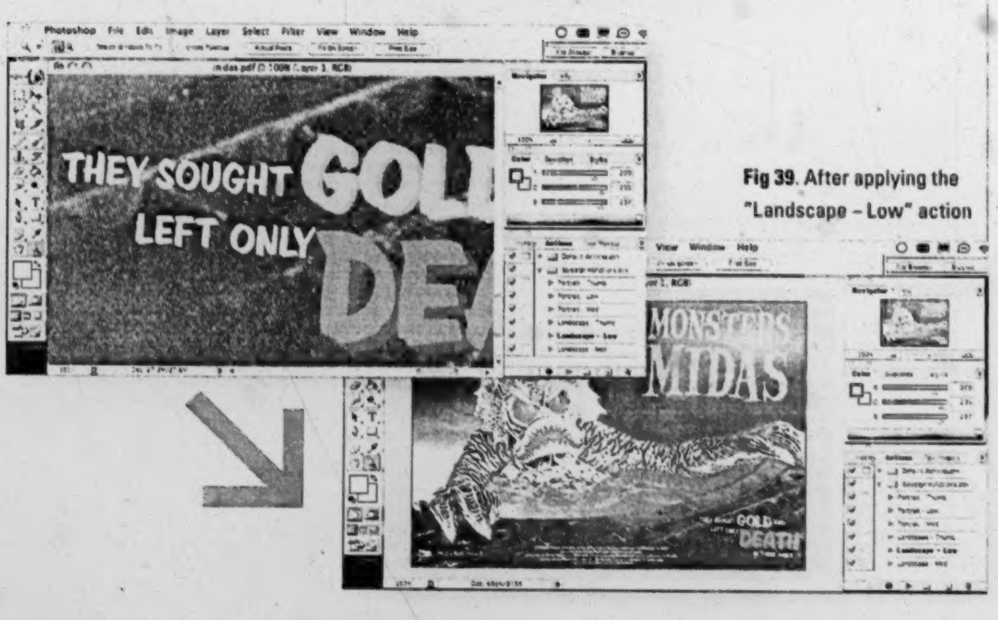

Photoshop File Edit Image Layer Select Filter View Window Help
THEY SOUGHT GOLD
LEFT ONLY DEA
MONSTERS
MIDAS
GOLD
DEATH
Actions
Portrait - Thumb
Portrait - Low
Portrait - Med
Landscape - Thumb
Landscape - Low
Landscape - Med

Fig 38. Before applying the "Landscape – Low" action

Fig 39. After applying the "Landscape – Low" action

Adding a Text Record

Now that we have prepared some images, it is finally time to return to our Streetprint site's administrative interface and add some text records. Streetprint's interface for managing your collection can be found by clicking the "My Collection" link on the administrative home page:

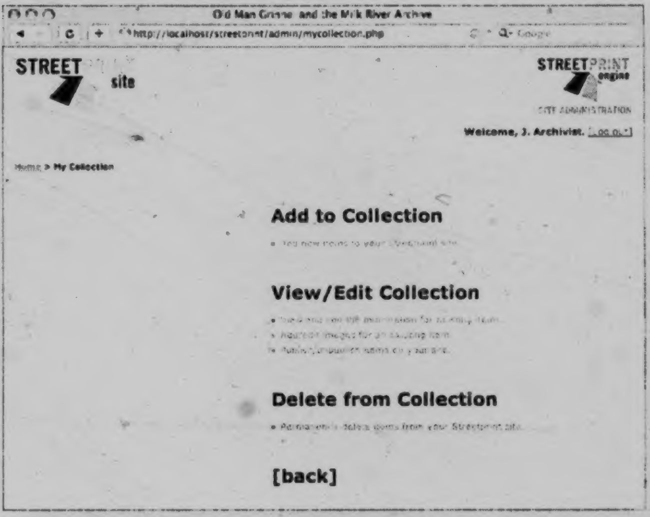

Old Man Grinne and the Milk River Archive
http://localhost/streetprint/admin/mycollection.php
STREET site
STREETPRINT engine
SITE ADMINISTRATION
Welcome, J. Archivist. [Log out]
Home > My Collection
Add to Collection
View/Edit Collection
Delete from Collection
[back]

Fig 34. Streetprint's collection management page

To add a collection record, click on "Add to Collection." You will see a page with a series of blank fields to fill in. The table below explains these fields in more detail. (Items in bold are suggestions from Team Streetprint.)

Fig. 35 Description of field names

Field Name	Required?	Description
Title	Yes	**A unique name for this collection item If the item is a book or text, simply place its title into this field For other items without obvious "titles," this is your chance to make up a name or phrase which will be unique to this item. (Note: two items can have the same title ONLY if they have different values in their "Author" fields.)**
Author	Yes	**The name of the author, artist, or organization responsible for the creation of this collection item. When listing proper names, use a "Lastname, Firstname" format**
Publisher	Optional	**This optional field can be used to store the name of the publisher (whether a person or a group) or printer associated with this collection item, if applicable.**

Field Name	Required?	Description
Document Type	Yes	This pull-down menu lists all of the document types that have been set up using Streetprint's "Type Management" features. Select the one which is most appropriate for this collection item.
Category	Optional	This menu lists all of the categories that have been set up using Street-print's "Type Management" features. By holding down a modifier key (Command or Shift on Macs, Control or Shift on PCs), you can select multiple categories for this item if appropriate.
Date (text)	Optional	This optional field is a text field which can be used to store the date associated with this collection item. This field is not actually used for sorting or ordering of any kind (see the notes for the 'Year' field below), and so you can use any convention for writing dates you like. **Maintain a consistent format between your records, however.**
Year (number)	Optional	This is a numerical field (numbers only) which stores the year associ-ated with this collection item. When a user browses your Streetprint site by date, this is the field used for sorting and ordering, although the "Date (text)" field will be displayed to the user. This implementation is a current weakness of the Streetprint Engine: sorting by year (as opposed to also sorting by month and day) is as "precise" as we can currently get, although whatever text is entered in the "Date (text)" field will always be displayed. We hope to improve Streetprint's handling of dates in the near future.
Dimensions	Optional	This optional text field is intended to store the physical measurements of this collection item. Though any text can be put in this field, decide on a standard unit of measurement and format which remains con-sistent across all your records.
Pagination	Optional	This is a numerical field (numbers only) intended to store the number of pages for this collection item, if applicable.
City	Optional	Depending on your needs, this field can store the city/country of this item's origin, or the city where it is currently located
Illustrations	Optional	If this item contains illustrations or other pictoral matter of special note, information about them can be placed in this field.
Location	Optional	This text field can be used for call numbers (if the item is part of a library collection), or other information regarding the current location of the collection item.
Notes	Optional	An important "catch-all" text field, this field is the place for any com-ments, expertise, or information which does not fit into any of the above fields.

While every single item in your Streetprint collection will have these same input fields, a collection item's optional fields will not be displayed to visitors to your website unless they contain information. This gives you some flexibility when cataloguing many different kinds of items and documents: we would be able to use the "Pagination" field when adding a record for a multiple-page letter, for example, but could leave the same field blank when adding a collection record for a movie poster.

The field names in your Streetprint site can also be changed to better suit your collection, as the next page explains. For the purposes of this manual, however, we will continue using the "default" fields names which we have described above.

For Power Users:

Customizing Streetprint's Field Names

Perhaps more than any other aspect of the site, Streetprint's "field names" betray our roots in the study of street literature—author, pagination, publisher, and so on. For the Milk River Archive, these fields are not always ideal but can still be used to describe the many different kinds of items in Old Man Grinnel's collection.

What if someone wanted to use the Streetprint Engine to create a digital archive of paintings, however? In such a scenario, the "Author" field should really be an "Artist" field, a field which would behave exactly like the "Author" field but with a more appropriate name.

Streetprint provides a facility for renaming fields through the **sp_fieldnames.inc file**, found inside your Streetprint directory. This file can be opened with any text editor (such as "TextEdit" on Macs or "Notepad" on Windows), and contains many lines which look like this:

```
$spFields['author'] = "Author";
```

By changing the words in double quotation marks, you can effectively re-label the "Author" field with any name you like. For our Streetprint painting site example, you would change the line to the following, and then save your changes to the **sp_fieldnames.inc file**:

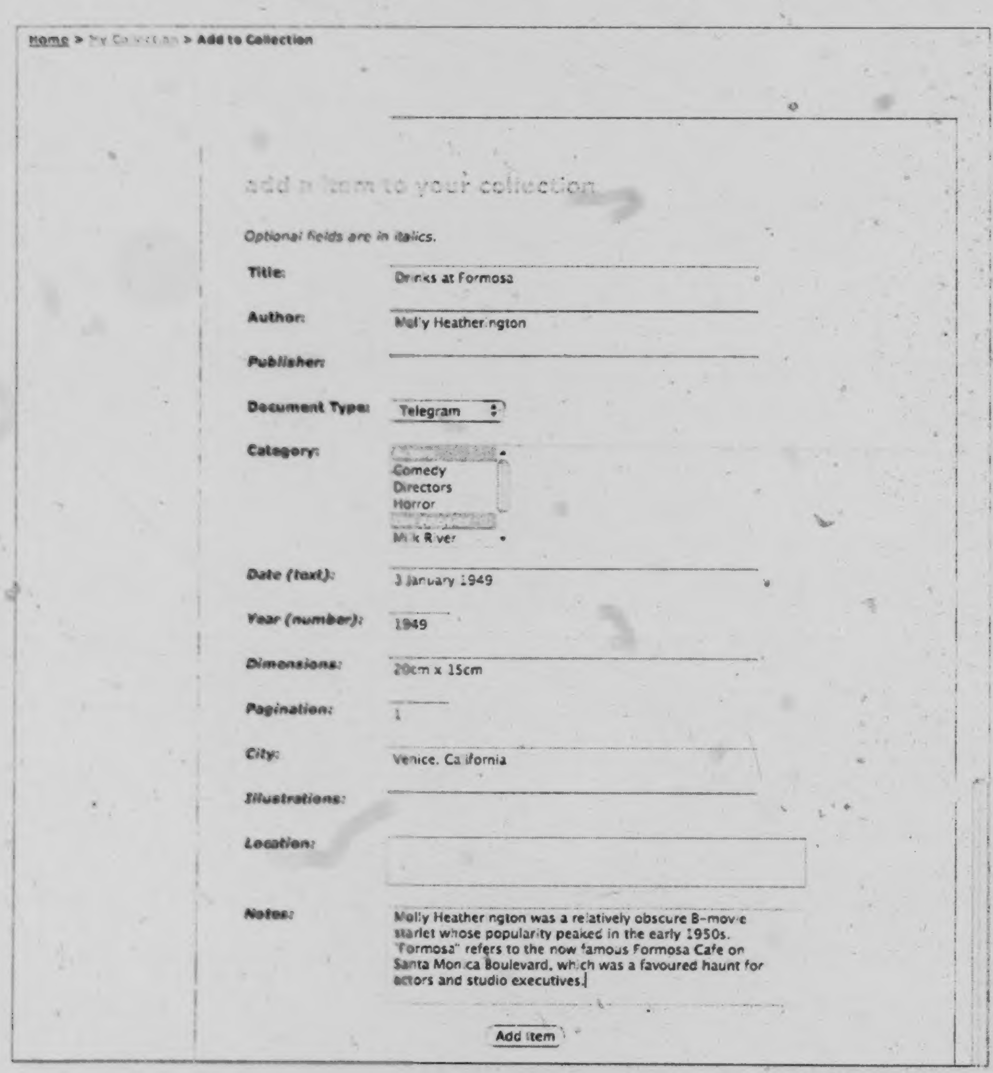
```
$spFields['author'] = "Artist";
```

It is important to remember that renaming a field does not change a field's function and behaviour—number fields remain number fields, optional fields remain optional, etc. Future versions of the Streetprint

Engine will gradually introduce more and more field customization options.

Let's add some items to our Streetprint site! We'll begin with the telegram from Molly Heatherington which we prepared earlier in the "Image Processing" section:

Fig 36. 'Add to Collection'



Home > My Collection > Add to Collection

add an item to your collection

Optional fields are in italics.

Title:	Drinks at Formosa
Author:	Molly Heatherington
Publisher:	
Document Type:	Telegram
Category:	Comedy Directors Horror Milk River
Date (text):	3 January 1949
Year (number):	1949
Dimensions:	20cm x 15cm
Pagination:	1
City:	Venice, California
Illustrations:	
Location:	
Notes:	Molly Heatherington was a relatively obscure B-movie starlet whose popularity peaked in the early 1950s. "Formosa" refers to the now famous Formosa Cafe on Santa Monica Boulevard, which was a favoured haunt for actors and studio executives.

Clicking the "Add" button will add the collection record to your Streetprint site. You will then be taken to the "Edit Details" page for your new item. From this page, you can view and edit all of this item's details, add and manage images, media files, and more.

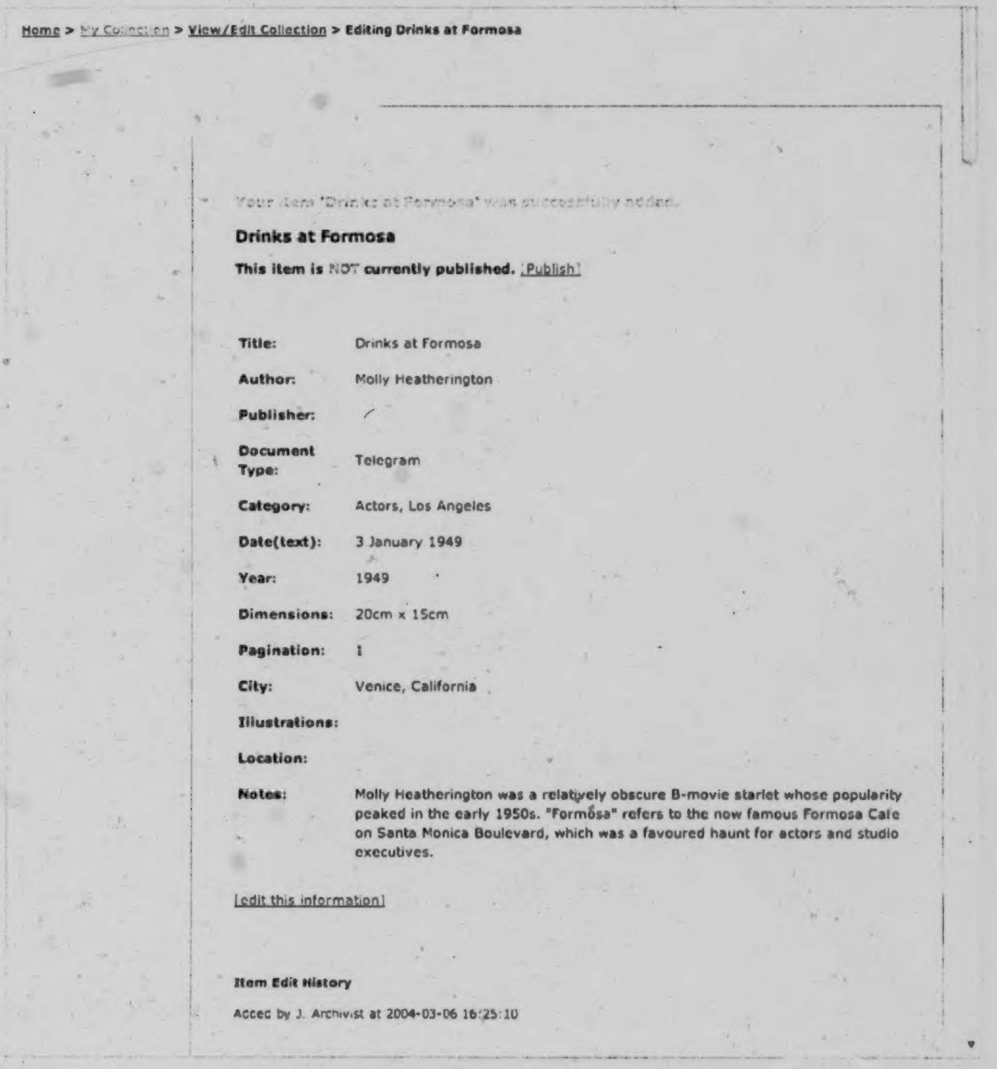


Fig 37. The first half of Streetprint's "Edit Details" page.

Home > My Collection > View/Edit Collection > **Editing Drinks at Formosa**

Your item 'Drinks at Formosa' was successfully added.

Drinks at Formosa

This item is NOT currently published. [Publish]

Title:	Drinks at Formosa
Author:	Molly Heatherington
Publisher:	
Document Type:	Telegram
Category:	Actors, Los Angeles
Date(text):	3 January 1949
Year:	1949
Dimensions:	20cm x 15cm
Pagination:	1
City:	Venice, California
Illustrations:	
Location:	
Notes:	Molly Heatherington was a relatively obscure B-movie starlet whose popularity peaked in the early 1950s. "Formosa" refers to the now famous Formosa Cafe on Santa Monica Boulevard, which was a favoured haunt for actors and studio executives.

[edit this information]

Item Edit History

Added by J. Archivist at 2004-03-06 16:25:10

Publish/Unpublish: Every item in your Streetprint collection is either published or unpublished. When you initially add an item to your collection, it is not published. While you can view and edit it through your site's administrative interface (as we are now doing with "Drinks at Formosa"), a visitor to www.oldmangrinnel.com would not be able to view this item or even be aware that it exists.

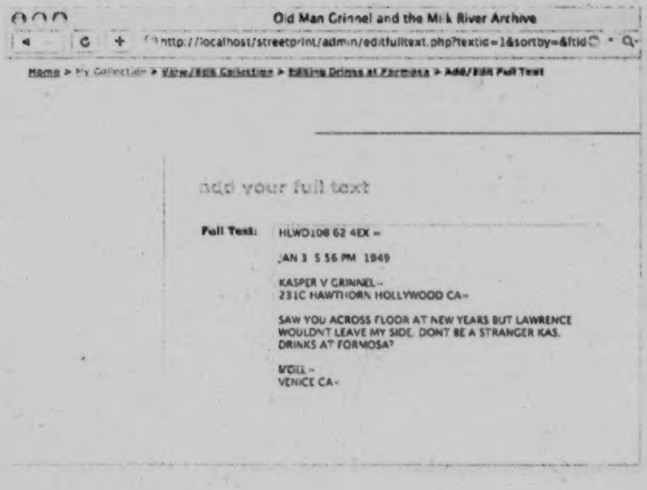
Once we have added images and our editorial judgement confirms that the item record is "ready to go," this item can be put online for the world to see by clicking the "Publish" link. Should a record need revision at any time, you can likewise "Unpublish" any currently-published collection item.

Item Details: All of the information we entered in the fields on the "Add to Collection" screen appear here. This information can be edited at any time by clicking the "Edit This Information" link.

Record History: Every time the information in the "Item Details" section is modified, a record is kept here. This is particularly useful if you are collaborating with others on your Streetprint site.

Image Management: Since images are central to any Streetprint collection, the image management section consists of several important screens. The next section of this chapter will cover them in detail.

Fulltext: If your collection item is of a textual nature—such as our telegram—you can optionally add the "full text" of the item to your collection record by clicking the "Add Fulltext" link. Users of your Streetprint site will then be able to search for this text and browse it alongside this item's images. The fulltext input box also accepts some basic XHTML markup tags if you want to add features such as bold, italics, or hyperlinks to your fulltext record.


Old Man Grinnel and the Milk River Archive
http://localhost/streetprint/admin/editfulltext.php?textid=1&sortby=&ftid
Home > My Collection > View/Edit Collection > Editing Drinks at Formosa > Add/Edit Full Text
add your full text

Full Text:

HLWD106 62 4EX –

JAN 3 5 56 PM 1949

KASPER V GRINNEL –
231C HAWTHORN HOLLYWOOD CA –

SAW YOU ACROSS FLOOR AT NEW YEARS BUT LAWRENCE WOULDNT LEAVE MY SIDE. DONT BE A STRANGER KAS. DRINKS AT FORMOSA?

MOLL –
VENICE CA –

Fig 38. Adding a fulltext record for the "Drinks at Formosa" telegram.

Media Management: In the media management section, you can add media files—such as movie clips, audio files, Flash movies, and more—to your collection record. To add a media file, simply click the "Choose File" button and select the media file from your computer's hard drive. Then choose the appropriate media type from the provided pull-down menu and optionally enter a textual description of the media file. Clicking "Add Media File" will then upload the file to your Streetprint site. Visitors to this collection item will see a "Media" icon which, when clicked, displays a popup window linking to all the media files associated with this item.

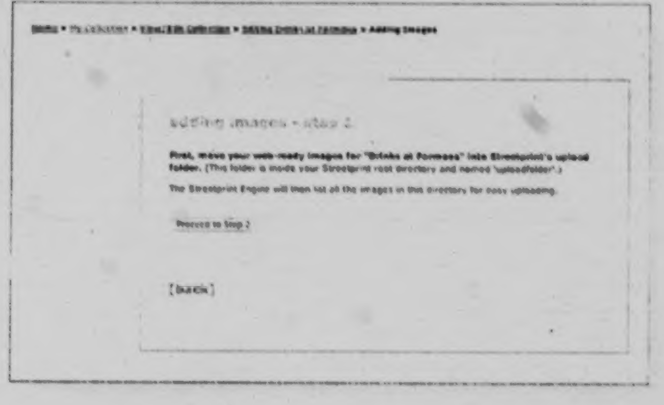
For more information on Streetprint's media file functionality, see the "Media Types" manual section on Page 26.

First lines: This feature of Streetprint was aimed at our founding collections of street ballads and other popular literature. Since these ballads often had no titles, or have titles which varied from printing to printing, it was customary to record the first lines of their texts as a method for cataloguing them.

While we won't need this feature for the Milk River Archive, using it is easy: type the line of text into the provided input box, and click the "Add First Line" button.

Adding Images

To add images to this collection record, click the "Add Images" link on this item's "Edit Details" page. You will then be prompted to move your web-ready images into Streetprint's *upload folder*:



adding images - step 1

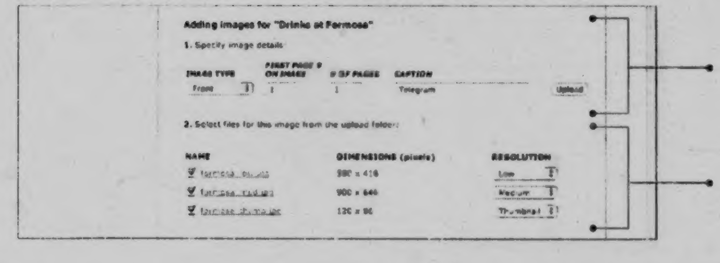
First, move your web-ready images for "Drinks at Formosa" into Streetprint's upload folder. (This folder is inside your Streetprint root directory and named 'uploadfolder'.)

The Streetprint Engine will then list all the images in this directory for easy uploading.

Fig 39. Adding images, step 1: move them into your site's upload folder

The second part of the screen lists all of the images in the upload folder. In our current example, our folder only contains three files, all of which are different resolutions of the "single" image for our telegram.

Let's enter the information for our telegram's image:



Adding images for "Drinks at Formosa"

1. Specify image details

IMAGE TYPE	FIRST PAGE # ON IMAGE	# OF PAGES	CAPTION
Front	1	1	Telegram

2. Select files for this image from the upload folder:

NAME	DIMENSIONS (pixels)	RESOLUTION
☑ [illegible]	980 x 416	Low
☑ [illegible]	900 x 644	Medium
☑ [illegible]	120 x 96	Thumbnail

Details of our first image

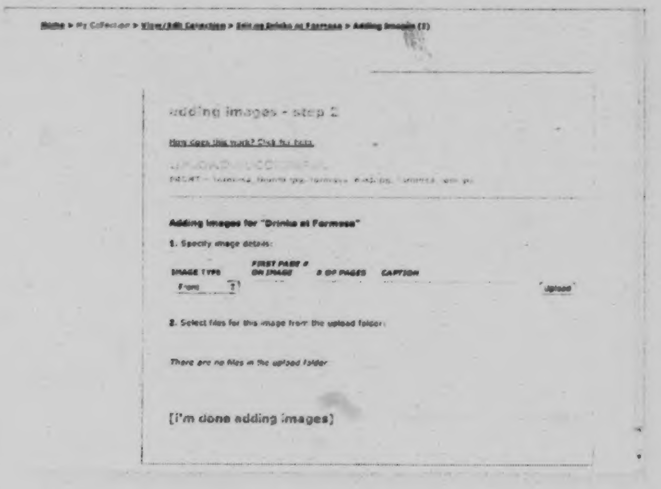
The files which "belong" to our first image

Fig 40. Specifying details and selecting files

Since we only have one image for this item, we check all three of our checkboxes—representing our three resolutions—and set the "Resolution" menus to indicate which file is which. (If you want to make sure you have the correct image, you can click the image name to see it in a new window.) When you have more than one image in your upload folder, Streetprint "learns" from your resolution menu settings and applies them to the remaining files, so you don't have to set the resolution menus every time.

Once we've specified our image details and resolutions, we're ready to click the upload button:

Fig 42. A successful upload



Home > My Collection > View/Edit Collection > Editing Drinks at Formosa > Adding Images (1)

adding images - step 2

How does this work? Click for help.

UPLOAD SUCCESSFUL

[illegible]

Adding images for "Drinks at Formosa"

1. Specify image details:

IMAGE TYPE	FIRST PAGE # ON IMAGE	# OF PAGES	CAPTION
Front			

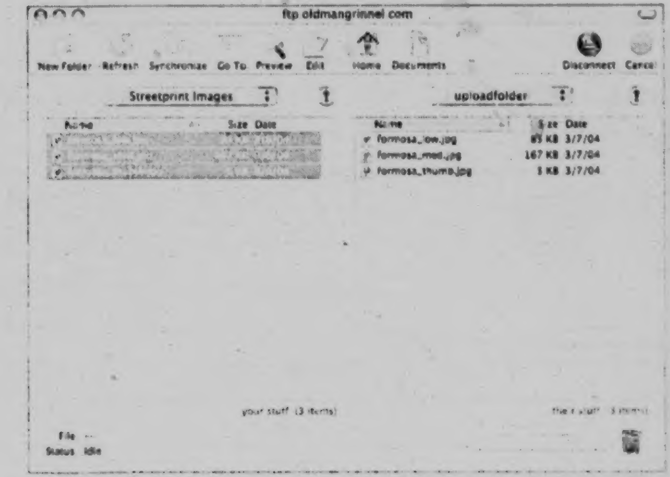
2. Select files for this image from the upload folder:

There are no files in the upload folder

[i'm done adding images]

This means you need to connect to your Streetprint web server—via a LAN connection (such as Appletalk) or via protocols such as FTP or scp—and copy the JPG files we prepared earlier for this item into the directory named uploadfolder, which is located inside your main Streetprint directory. (If an internet geek or a support staff member helped you install your Streetprint site, you may wish to consult with them again to help you through this process.) Since you will be moving images into your upload folder fairly often, it's a good idea to place a shortcut to this location on your computer.

Fig 43. Placing images in the upload folder using an FTP client

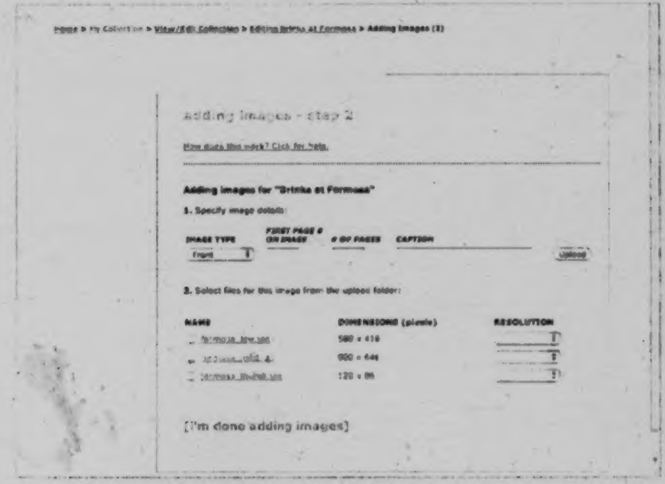

ftp oldmangrinnel com
New Folder
Refresh
Synchronize
Go To
Preview
Edit
Home
Documents
Disconnect
Cancel
Streetprint Images
uploadfolder
your stuff (3 items)
File --
Status Idle

Name	Size	Date
formosa_low.jpg	85 KB	3/7/04
formosa_med.jpg	167 KB	3/7/04
formosa_thumb.jpg	3 KB	3/7/04

Once your images have been uploaded, click the "Proceed to Step 2" button to proceed to the next screen:

This screen consists of two parts. The first allows you to specify the details of current image you are adding: the image type, the first page number which appears on the image (if any), the total number of pages represented in the image (if any), and a caption for this image. Only the image type menu is mandatory.

Fig 44. Adding images Step 2: specifying details and selecting


Home > My Collection > View/Edit Collection > Editing Britka At Formosa > Adding Images (2)
adding images - step 2
How does this work? Click for help.
IMAGE TYPE
FIRST PAGE # ON IMAGE
# OF PAGES
CAPTION
Front
Upload
RESOLUTION
[I'm done adding images]

Adding images for "Britka at Formosa"

1. Specify image details:

2. Select files for this image from the upload folder:

NAME	DIMENSIONS (pixels)
formosa_low.jpg	580 × 416
formosa_med.jpg	800 × 644
formosa_thumb.jpg	120 × 86

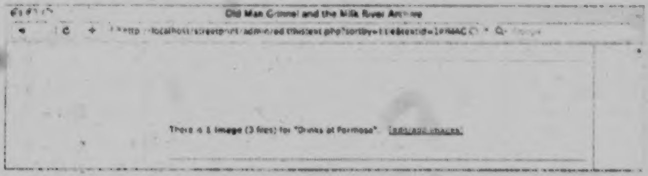
Our image for "Drinks at Formosa" has been successfully uploaded, our upload folder is now empty, and we can safely click the "I'm done adding images" link. If we were adding images for a collection item with multi-ple pages of images, however, we would simply continue specifying image details (and selecting the appropriate file checkboxes) until the upload folder is emptied.

Althrough three of the "image details" input fields are optional—"First Page # on Image," "# of Pages," and "Caption"—you can use them to your advantage if you know a bit more about how the Streetprint Engine works.

If you enter text into the "Caption" field, your caption will always be displayed to website visitors beneath your image. If this field is left empty, Streetprint looks instead to the page-related fields and uses that information—if it exists—to create captions such as "Page 1" or "Pages 32-33." If all these fields are blank, Streetprint uses the image type as the caption.

Streetprint also uses the "First Page # on Image" field for something else: the order in which it displays a collection item's images. If you type in a caption for an image, the value of the "First Page #" field will never be seen by your visitors, since the caption will always be displayed instead. This allows you to use the page number field as an "ordering" field if you so choose. The field can even accept negative numbers; if you're entering a book which has a cover and a prefatory page before "Page 1," you can assign them page numbers of -1 and 0, respectively. These numbers will not be seen as long as captions (such as "Cover" and "Preface") are provided. Once you reach the Page 1 image, however, you can leave the caption field blank and Streetprint will automatically create the "Page X" captions for you.

Fig 45. The "Edit Details" screen after adding an image

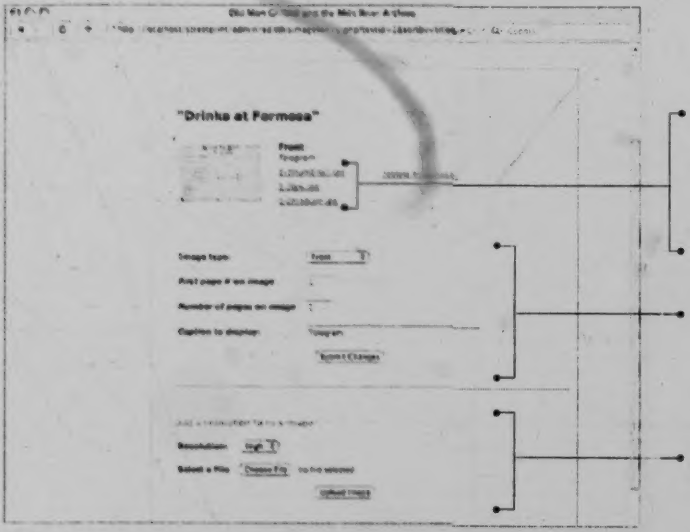


There is 1 image (3 files) for "Drinks at Formosa". [edit/add images]

Clicking "I'm Done Adding Images" will return us to our collection item's "Edit Details" screen.

To add more images to this item, or to edit our current images (and associated information, such as captions), we can simply click the "edit/add images" link on this page.

Fig 46. Editing an Image



"Drinks at Formosa"

Front
Telegram

Image type: Front
First page # on image: 1
Number of pages on image: 1
Caption to display: Telegram

Click a filename to view a resolution image, and to optionally swap this resolution with another image file

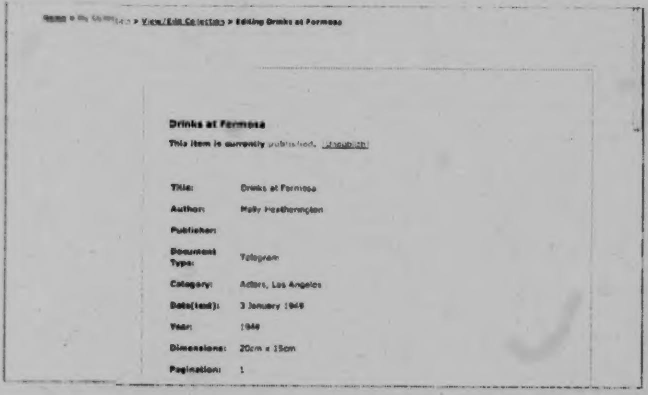
Edit image details

Individually add additional resolutions

Publishing Your Item

Now that we have a complete item record along with images, we can safely put this item online for the world to enjoy by clicking the "Publish" link on the "Edit Details" page:

Fig 47. A published collection item



View/Edit Collection > Editing Drinks at Formosa

Drinks at Formosa

This item is currently published. [Unpublish]

Title: Drinks at Formosa
Author: Molly Heatherington
Publisher:
Document Type: Telegram
Category: Actors, Los Angeles
Date(text): 3 January 1946
Year: 1946
Dimensions: 20cm x 15cm
Pagination: 1

Since this item is currently the only item in our new collection, we may as well make it the "Featured Item" on our website!

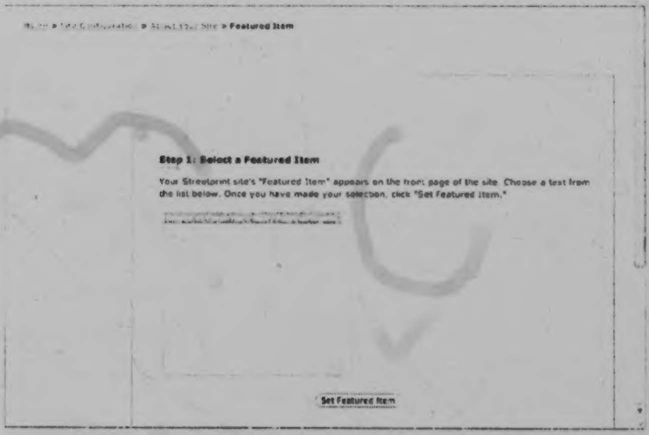


Fig 48. Setting our featured item

▸ Featured Item

Step 1: Select a Featured Item

Your Streetprint site's "Featured Item" appears on the front page of the site. Choose a text from the list below. Once you have made your selection, click "Set Featured Item."

Set Featured Item

A visitor to www.oldmangrinnel.com would now see this:

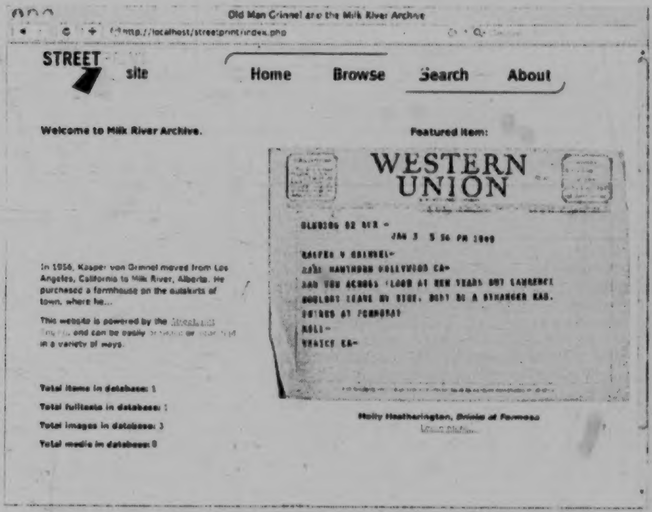


Fig 49. Front page of the Milk River Archive Streetprint site

Old Man Grinnel and the Milk River Archive
http://localhost/streetprint/index.php
STREET
site
Home
Browse
Search
About
WESTERN
UNION

Welcome to Milk River Archive.

In 1956, Kasper von Grinnel moved from Los Angeles, California to Milk River, Alberta. He purchased a farmhouse on the outskirts of town, where he...

This website is powered by the [illegible] and can be easily [illegible] in a variety of ways.

Total items in database: 1
Total fulltexts in database: 1
Total images in database: 3
Total media in database: 0

Featured Item:

Holly Heatherington, *Brinie of Formosa*

Here is how the record for "Drinks at Formosa" appears to a website visitor:

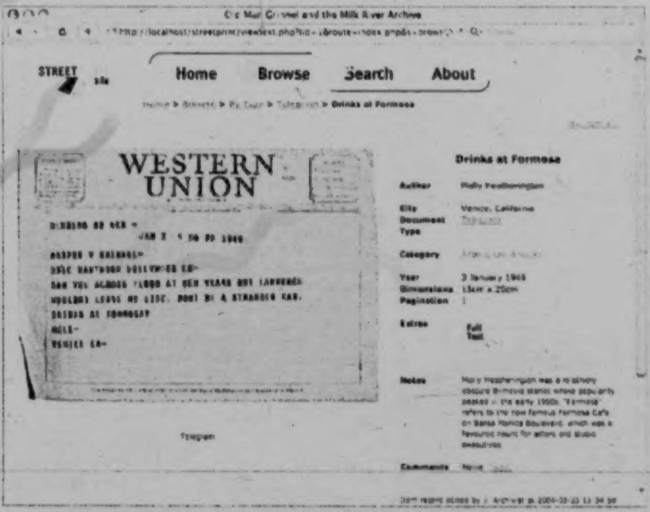

Home
Browse
Search
About
WESTERN UNION
Drinks at Formosa

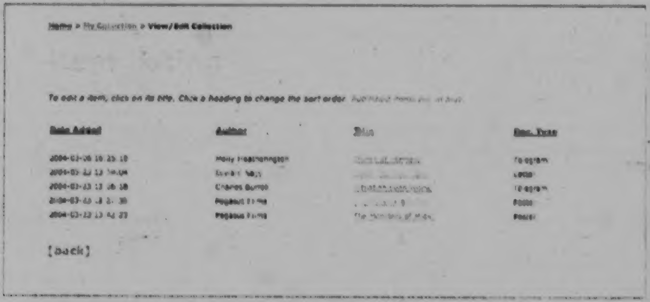
Fig 50. Viewing "Drinks at Formosa" online

Editing Your Collection

Editing your collection items is easy: simply click on "View/Edit Collection" from Streetprint's collection management page. You will be presented with a list of your collection items, colour-coded by publishing status. You can also click on any column title to sort the list:

If you are an "Editor" user, you can click on any title to access that item's text details screen. "Data Entry" users will be able to click on the titles of items which they personally have added to the collection.

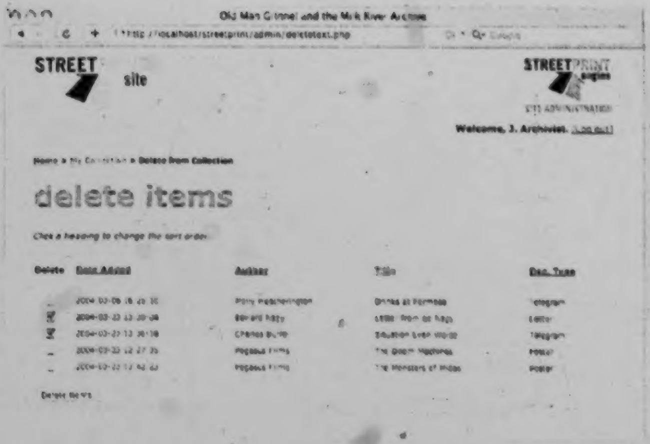
Fig 51. Collection listing sorted by Title


View/Edit Collection
[back]

Deleting From Your Collection

Only delete items from your collection which you are sure you do not want; in most cases, unpublishing a collection item is a better option. To delete items, select "Delete from Collection" from Streetprint's collection management page. Check boxes indicating the items you wish to delete, and click the "Delete Items" button.

You will be prompted to confirm your choices before the items are deleted.

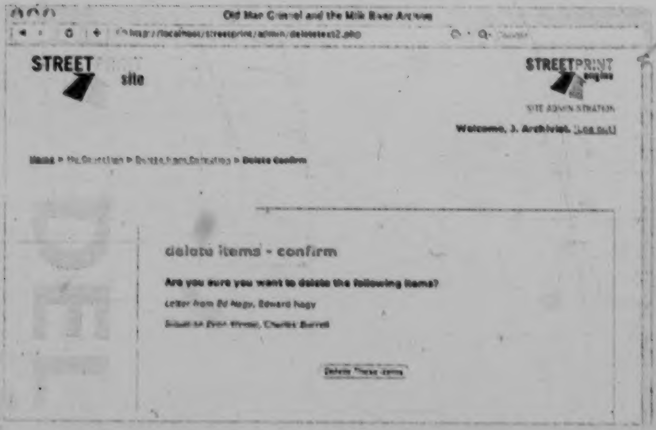


STREET
site
STREETPRINT engine
delete items

Remember to use this feature with caution: deleted collection items (and their associated images, media files, etc.) cannot be recovered.

Fig 53. Deleting collection items

Fig 52. Are you really really sure?



STREET
site
STREETPRINT engine
delete items - confirm

Are you sure you want to delete the following items?

Letter from Ed Nagy, Edward Nagy

That's All, Folks

We've now learnt everything we need to know to add more items to the Milk River archive! If you're curious to see more items from Old Man Grinnel's collection, you can view the website we've been building during this manual online at: *www.streetprint.org/oldmangrinnel/*

The collection contains items with media files, multiple images, fulltext records, and more.

Our Streetprint site is missing just one thing: a graphical "skin" which gives the site a unique look and identity suited to the collection. Skins are the topic of the next chapter, so read on...

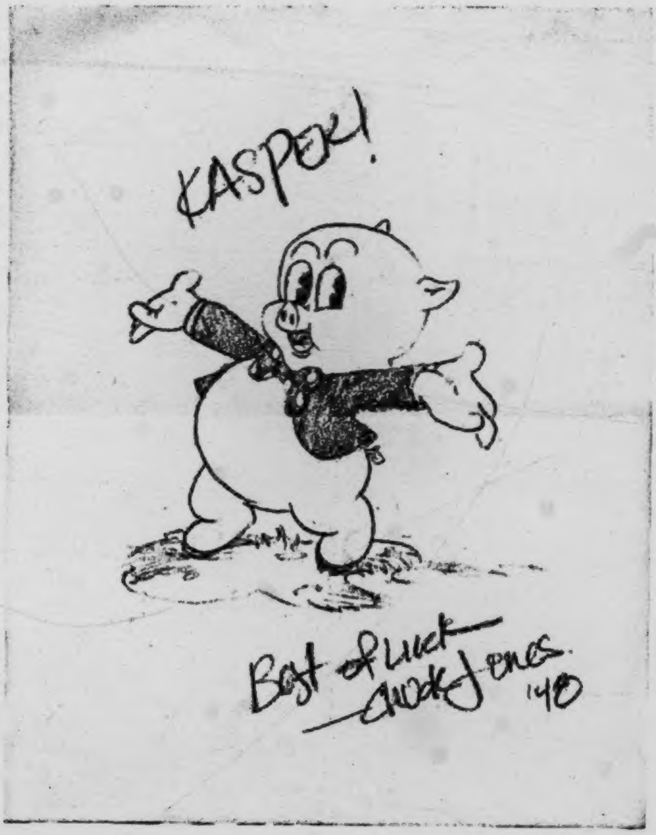


Fig 54. Autograph from Charles M. Jones 1948

CHAPTER V

SKINS

Comfortable In Your Skin

Every Streetprint collection is unique. The various websites at Streetprint.org currently showcase everything from 18th-century chapbooks *(Revolution & Romanticism)* to rave posters *(Urban Record)* to clothing and cultural artifacts *(Eritrean Print and Oral Culture)*.

Since each collection is unique, Team Streetprint realized that every Streetprint site should also look unique. The team accomplished this by making it very easy to change the graphics which make up your site's visual look and theme, or "skin."

The Streetprint Engine comes with a default skin which we have been using thus far for Old Man Grinnel's collection. But we hardly want the public to visit the Milk River Archive's website and only see the generic "Streetprint Site" logo in the corner. It's time to customize this skin to suit our collection; this chapter will show you how.

For Power Users:

Streetprint Skins and Web Standards

In the two years since the Streetprint Engine was first developed, the web has seen a tremendous grassroots push for the increased adoption of standards, with a particular emphasis on encapsulating a site's layout and overall design in its CSS (Cascading Style Sheet) file(s). This approach has clear benefits; with proper XHTML and CSS markup, an entire website can receive a visual makeover simply by tweaking its stylesheet.

The current version of the Streetprint Engine still uses images for its primary navigation (the graphics for 'Home,' 'Browse,' etc.) and its stylesheet (sp_style.css) provides only rudimentary control over the look of a Streetprint site. The next major revision of the Engine, due in late 2004, will drastically change this. We plan to move to a nearly all-CSS layout, which means that "skinning" a Streetprint site will become much easier (and more standards compliant). We will also provide backwards compatibility for existing sites, and for users who prefer using images for their navigation menu.

The Graphics Directory

All of the graphics which make up a site's skin are located in a directory called *gfx/* located inside our main Streetprint directory. This directory contains an image file for every "interface element" in a Streetprint site. Have a look at the image names and the graphics themselves; in most cases, it's easy to guess what an image is for.

Fig 55. logotype.gif

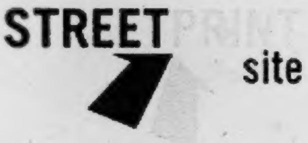


Fig 56. search.gif

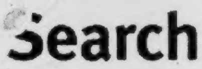


Fig 57. resolution_low.gif, resolution_med.gif, resolution_high.gif

The graphics which make up a Streetprint site's navigation bar are slightly more complicated. Every element in the navigational bar—such as *Home* or *Browse*—can have up to three "states," each represented by a slightly different image. In addition to the default state (or "nostate"), a navigation graphic may have a "roll state" (the image shown when a user rolls their mouse over the graphic) and an "on state" (the image shown when a user is currently within that particular section of the site). Some site editors prefer to only have two states, making the "on" and "roll" states one and the same, whereas others prefer to use all three. Streetprint permits both configurations.

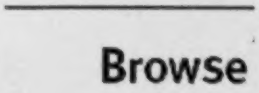


Fig 58. browse_nostate.gif
The default state

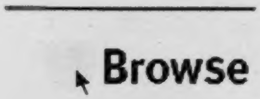


Fig 59. browse_rollstate.gif
The "roll over" state

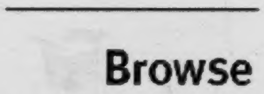


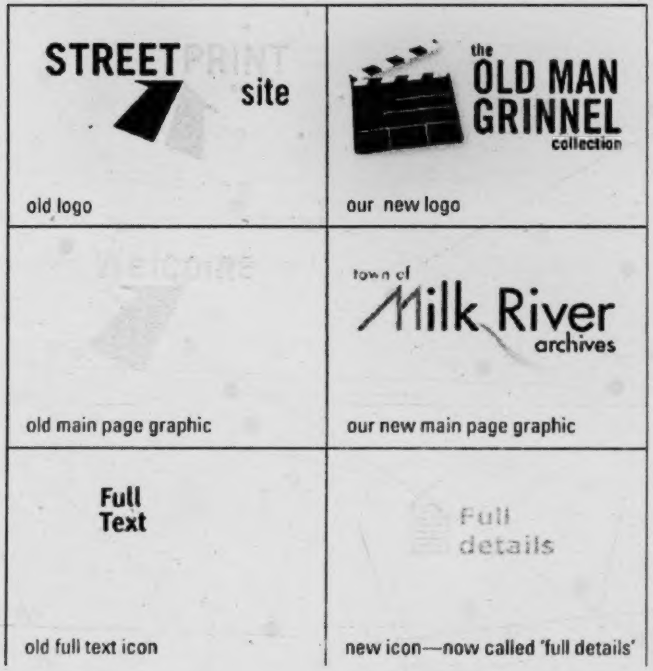
Fig 60. browse_onstate.gif
The "on" state, we're now in the 'browse' section

Roll Your Own

Now that we've found all the necessary graphics, it's time to replace them with our own! The default skin provides a good set of guidelines for the size of our images, but we are free to make any image slightly larger or smaller than the file it will replace. As well, Streetprint allows us to use JPEGs instead of GIFs if we so choose.

For Old Man Grinnel's collection, we decided to make some simple changes, incorporating a new background colour, typeface, and a slight Hollywood aesthetic. (You can make your own graphics files in Adobe® Photoshop® or any other program which allows you to edit and save GIFs or JPEGs.) Here's a look at some of the graphics in our new skin:

Fig 61. Old and new graphics


STREETPRINT
site
old logo
the
OLD MAN
GRINNEL
collection
our new logo
Welcome
old main page graphic
town of
Milk River
archives
our new main page graphic
Full
Text
old full text icon
Full
details
new icon—now called 'full details'

Should we need to change any of our image file names—if we are using JPEGs instead of GIFs, for example, or if we want to use all three navigation states instead of two—the necessary adjustments can be made in the sp_config.php file in our main Streetprint directory.

The sp_config.php file can be opened with any text editor, such as TextEdit on a Mac or Notepad on Windows. This file's "Graphics Variables" section contains a series of statements which map a specific Streetprint graphic, such as "logo," to a specific filename, such as "logotype.gif."

Let's say that we didn't like Streetprint's convention of using two different sizes of logo (one for the main page and one for other pages), and we wanted to use a single file named "logo.jpg" instead. This could be accomplished by changing the lines which read

```
$graphics['logo'] = "logotype_small.gif";
$graphics['logobig'] = "logotype.gif";
```

to:

```
$graphics['logo'] = "logo.jpg";
$graphics['logobig'] = "logo.jpg";
```

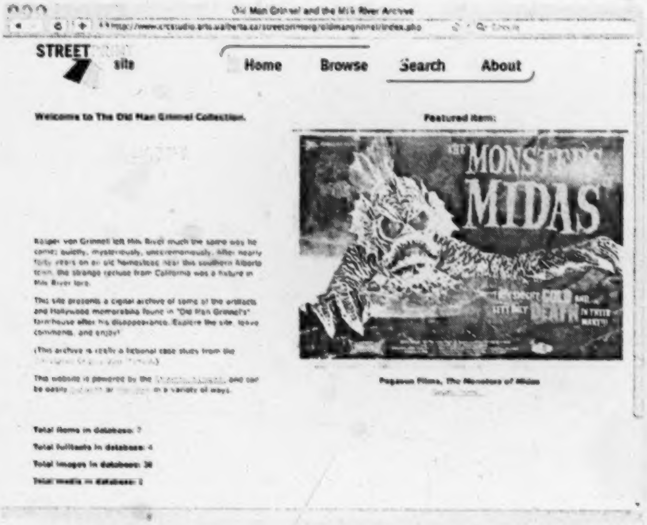
The rest of the graphics variables section can be modified in a similar fashion. For the Old Man Grinnel website, we are using GIF files and haven't changed any of their names, and so no modification of sp_config.php is required.

Finishing Up

Once we've moved our new graphics into the *gfx/* directory and made tweaks to *sp_config.php* (if neccessary) , our skin is ready to go! If we want to change our site's typeface, background colour or image, or link colour, however, we can do so by editing our site's stylesheet. The stylesheet is located in the main Streetprint directory and named *sp_style.css*. Once you learn a few basics, editing CSS stylesheets is extremely easy. A good introduction to CSS is on the web at: *www.yourhtmlsource.com/stylesheets/introduction.html*

Here's how our final skin looks on the Old Man Grinnel website, which can be found online at www.streetprint.org/oldmangrinnel:

Fig 61. The old site


Old Man Grinnel and the Milk River Archive
STREET
site
Home
Browse
Search
About

Welcome to The Old Man Grinnel Collection.

Featured Item:

Total items in database: 7

Total fulltexts in database: 4

Fig 62. Our new site!

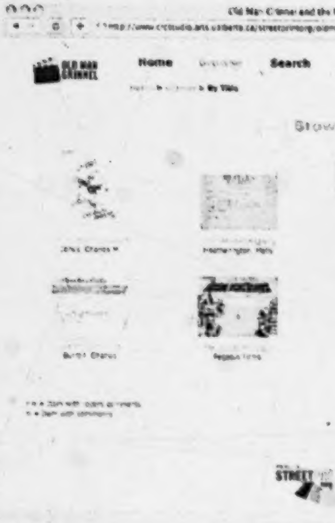
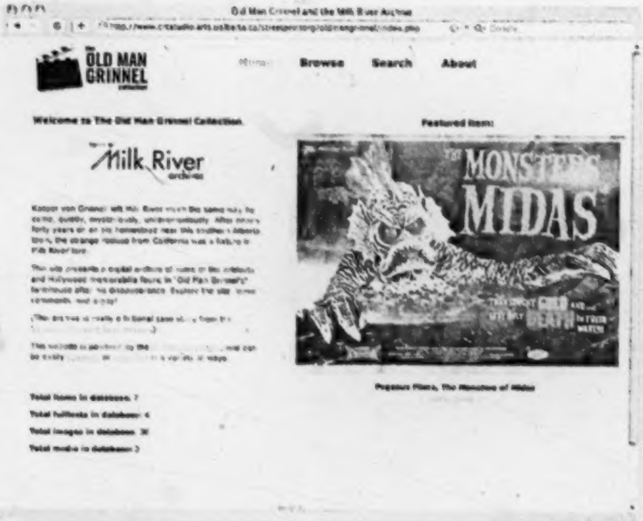

Old Man Grinnel and the Milk River Archive
OLD MAN GRINNEL
Home
Browse
Search
About
Browsing

Fig 63. The old man would be proud!


Old Man Grinnel and the Milk River Archive
OLD MAN GRINNEL
Home
Browse
Search
About
Milk River

Welcome to The Old Man Grinnel Collection.

Featured Item:

Welcome Aboard

That's it! Our new Streetprint site is now ready for the world.

Every Streetprint collection is unique, and so adding your own skin and tweaking your style sheet might sometimes be difficult. You may have questions to ask about the skinning process, or about other aspects of your Streetprint site. Even if everything has gone smoothly, you might simply want to let everyone know about your cool new website. Either way, there's a perfect place to start: *the Streetprint.org forums!* Visit them online at:

www.streetprint.org/forums/

Team Streetprint regularly checks in, and we'd love to help you set up your site.

By powering your digital collection with the Streetprint Engine, you're joining a cutting-edge, evolving community of open archives. We're excited to have you aboard. Don't be afraid to stop in and say hello!